BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI TP. HỒ CHÍ MINH
BỘ MÔN CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN TỐT NGHIỆP

ĐỀ TÀI: NGHIÊN CỨU THUẬT TOÁN K-NEAREST NEIGHBORS VÀ ỨNG DỤNG VÀO BÀI TOÁN DỰ ĐOÁN TÌNH TRẠNG HÔN NHÂN

Giảng viên hướng dẫn: TRẦN PHONG NHÃ

Sinh viên thực hiện: TRẦN XUÂN LÂM

Lớp: CÔNG NGHỆ THÔNG TIN

Khoá: 57

TP. Hồ Chí Minh, tháng 08 năm 2020

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI TP. HỒ CHÍ MINH
BỘ MÔN CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN TỐT NGHIỆP

ĐỀ TÀI: NGHIÊN CỨU THUẬT TOÁN K-NEAREST NEIGHBORS VÀ ỨNG DỤNG VÀO BÀI TOÁN DỰ ĐOÁN TÌNH TRẠNG HÔN NHÂN

Giảng viên hướng dẫn: TRẦN PHONG NHÃ

Sinh viên thực hiện: TRẦN XUÂN LÂM

Lớp: CÔNG NGHỆ THÔNG TIN

Khoá: 57

TP. Hồ Chí Minh, tháng 08 năm 2020

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH

CỘNG HÒA XÃ HỘI CHỦ NGHIÃ VIỆT NAM
Độc lập – Tự do – Hạnh phúc

NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP

BỘ MÔN: **CÔNG NGHỆ THÔNG TIN**

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

Mã sinh viên: 5751071021 **Họ tên SV:** Trần Xuân Lâm

Khóa: 57 **Lớp:** Công Nghệ Thông Tin

1. Tên đề tài.

NGHIÊN CỨU THUẬT TOÁN K-NEAREST NEIGHBORS VÀ ỨNG DỤNG VÀO BÀI TOÁN DỰ ĐOÁN TÌNH TRẠNG HÔN NHÂN.

2. Mục đích, yêu cầu.

a. Mục đích.

Xây dựng một công cụ dự đoán, dự báo nhằm phục vụ cho các cá nhân, tổ chức, doanh nghiệp có nhu cầu trong việc xác định tình trạng hôn nhân.

b. Yêu cầu.

- Tìm hiểu về Machine Learning.
- Nghiên cứu về thuật toán K-Nearest Neighbors.
- Xây dựng, thiết kế, đánh giá thuật toán dựa trên tập dữ liệu mẫu.
- Ứng dụng thuật toán vào bài toán dự đoán tình trạng hôn nhân.

3. Nội dung và phạm vi đề tài.

a. Nội dung đề tài.

- Giới thiệu tổng quan về trí tuệ nhân tạo và Machine Learning.
- Giới thiệu về thuật toán K-Nearest Neighbors.
- Nghiên cứu, phân tích, đánh giá thuật toán K-Nearest Neighbors.
- Ứng dụng thuật toán vào bài toán thực tế, cụ thể là bài toán dự đoán tình trạng hôn nhân.

b. Phạm vi đề tài.

- Nghiên cứu thuật toán K-Nearest Neighbors và ứng dụng thuật toán vào dự đoán tình trạng hôn nhân.

4. Công nghệ, công cụ và ngôn ngữ lập trình.

a. Công nghệ: HTML, CSS, Javascript, Bootstrap.

b. Ngôn ngữ lập trình: Sublime Text phiên bản 3.2.2.

5. Các kết quả chính dự kiến sẽ đạt được và ứng dụng

- Hoàn chỉnh cuốn báo cáo đề tài.
- Khái quát được tổng quan về Machine Learning.
- Nắm được thuật toán K-Nearest Neighbors và có thể áp dụng được thuật toán cho bất kỳ bài toán nào liên quan.
- Nắm được các ưu, nhược điểm của thuật toán, phương pháp tối ưu cho thuật toán.
- Xây dựng được website demo dự doán tình trạng hôn nhân.

6. Giáo viên và cán bộ hướng dẫn

Họ tên: TRẦN PHONG NHÃ

Đơn vị công tác: Bộ môn Công Nghệ Thông Tin – Trường Đại học Giao thông Vận tải phân hiệu tại TP HCM

Điện thoại: 0906761014 Email: tpnha@utc2.edu.vn

Ngày … tháng … năm 2020	**Đã giao nhiệm vụ TKTN**
BM Công Nghệ Thông Tin	**Giáo viên hướng dẫn**

Trần Phong Nhã

Đã nhận nhiệm vụ TKTN

Sinh viên: Trần Xuân Lâm Ký tên:

Điện thoại: 0363131199 Email: trxlamit@gmail.com

LỜI CẢM ƠN

Lời nói đầu tiên, tôi xin kính gửi lời cảm ơn chân thành nhất tới Quý thầy cô trong Bộ môn Công Nghệ Thông Tin, cũng như Ban Giám Hiệu Trường Đại học Giao thông Vận tải phân hiệu tại Thành phố Hồ Chí Minh, đã cho phép tôi thực hiện đề tài tốt nghiệp **"Nghiên cứu thuật toán K-Nearest Neighbors và Ứng dụng vào bài toán dự đoán tình trạng hôn nhân"**.

Để hoàn thành nhiệm vụ được giao này, ngoài sự nỗ lực học hỏi không ngừng của bản thân còn có sự hướng dẫn tận tình của các giảng viên trong 4 năm vừa qua, đặc biệt hơn hết nhờ có giảng viên **Trần Phong Nhã**, người thầy đã hướng dẫn cho tôi những hướng đi, truyền đạt cho tôi những kiến thức, kỹ năng để tôi có thể hoàn thành đề tài tốt nghiệp này.

Mặc dù đã cố gắng hết sức để hoàn thành đề tài, nhưng chắc chắn rằng sẽ khó tránh khỏi những thiếu sót. Tôi rất mong nhận được những sự đánh giá, góp ý của Quý thầy cô để tôi có thể rút ra cho mình những bài học, kinh nghiệm quý báu.

Sau cùng, tôi cũng không biết nói gì hơn ngoài kính chúc Quý thầy cô trong Bộ môn Công Nghệ Thông Tin và đặc biệt là thầy **Trần Phong Nhã** thật dồi dào sức khỏe và ngày càng gặt hái được nhiều thành công hơn nữa trong cuộc sống cũng như trong sự nghiệp giảng dạy của mình.

Tôi xin chân thành cảm ơn!

TP. Hồ Chí Minh, ngày … tháng … năm 2020

Sinh viên thực hiện

Trần Xuân Lâm

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

TP. Hồ Chí Minh, ngày … tháng … năm 2020

Giảng viên hướng dẫn

Trần Phong Nhã

MỤC LỤC

DANH MỤC THUẬT NGỮ

STT	TỪ VIẾT TẮT	TÊN TIẾNG ANH	Ý NGHĨA TIẾNG VIỆT
1	CNTT	Information Technology	Công Nghệ Thông Tin
2	SPAM	Stupid Pointless Annoying Messages	Thư rác
3	AI	Artificial Intelligence	Trí tuệ nhân tạo
4	ML	Machine Learning	Học máy/Máy học
5	DL	Deep Learning	Học sâu
6	CAD	Computer-Aided Design	Thiết kế dưới sự hỗ trợ của máy tính
7	GUI	Graphical User Interface	Giao diện đồ họa người dùng
8	Numpy	Nummerical Python	
9	HTML	HyperText Markup Language	Ngôn ngữ đánh dấu siêu văn bản
10	CSS	Cascading Style Sheets	
11	JS	JavaScript	
12	KNN	K-Nearest Neighbors	K láng giềng gần nhất
13	E	Experience	Kinh nghiệm
14	T	Task	Tác vụ
15	P	Performance	Hiệu suất
16	SL	Supervised Learning	Học có giám sát

DANH MỤC BẢNG BIỂU

DANH MỤC HÌNH ẢNH

TỔNG QUAN.

Tính cấp thiết của đề tài.

Những năm gần đây, cuộc cách mạng công nghiệp 4.0 (cách mạng công nghiệp lần thứ 4) đang nổi lên với sự phát triển của trí tuệ nhân tạo (Artificial Intelligence). Trí tuệ nhân tạo có thể được hình dung đơn giản là một bộ não bằng máy móc và biết suy nghĩ, thậm chí nó còn có cảm xúc, khủng khiếp hơn nó có thể còn có cả tham vọng quyền lực như trong các bộ phim viễn tưởng, nhưng đó chỉ là giả định như trong cuộc tranh luận giữa Jack Ma và Elon Musk.

AI còn được chia thành nhiều tập con khác, 'Học máy' hay 'Máy học' (Machine Learning) là 1 lĩnh vực của Khoa học máy tính và là 1 tập con của AI. Nó có khả năng tự học hỏi dựa trên dữ liệu đưa vào mà không cần phải được lập trình cụ thể. 'Học sâu' (Deep Learning) là 1 phần của ML.

Hầu hết mọi nghành công nghiệp đang làm việc với hàm lượng lớn dữ liệu đều nhận ra tầm quan trọng của công nghệ ML. Nhưng cái nhìn sáng suốt từ nguồn dữ liệu này – chủ yếu đang thời gian thực – sẽ giúp các tổ chức vận hành hiệu quả hơn hoặc tạo lợi thế cạnh tranh so với các đối thủ.

Xu hướng phát triển công nghệ thông tin ngày càng tăng, song song với nó lượng dữ liệu được sinh ra cũng ngày một lớn. Vì vậy nhu cầu để xử lý dữ liệu cũng lớn hơn, ML đang góp phần giải quyết vấn đề này. Một trong những thuật toán thường dùng trong ML đó là thuật toán K- nearest neighbor. Ứng dụng của thuật toán này được sử dụng rất nhiều và rộng rãi trong các bài toán phân lớp.

Song, trước sự tiến bộ của khoa học công nghệ nói riêng và sự tiên tiến của thời đại nói chung, thì ngày nay tình trạng ly hôn, các cuộc chia tay của các cặp gia đình, cặp đôi ngày càng nhiều. Vấn đề này không chỉ ảnh hưởng đến mối quan hệ trong các gia đình, mà còn ảnh hưởng đến cả toàn bộ xã hội. Các đứa trẻ mồ côi, thiếu cha hay thiếu mẹ đều có ảnh hưởng đến quá trình nhận thức và trưởng thành của chúng. Đây cũng là một phần lý do ảnh hưởng lớn đến việc các tệ nạn xuất hiện ngày càng nhiều ở giới trẻ.

Để hạn chế tình trạng ly hôn và tận dụng sự phát triển mạnh của ML, tôi quyết định thực hiện: **"Nghiên cứu thuật toán K-Nearest Neighbors trong Machine Learning và Ứng dụng vào bài toán dự đoán ly hôn"** để làm đề tài đồ án tốt nghiệp.

Mục tiêu nghiên cứu.

Sau khi đã xác định được đề tài, tôi đặt mục tiêu nghiên cứu các vấn đề sau:

- Đầu tiên, nghiên cứu về Machine Learning.
- Sau đó, nghiên cứu thuật toán K-Nearest Neighbors.
- Cuối cùng, sử dụng thuật toán đó để áp dụng vào trong dự đoán ly hôn.

Đối tượng và phạm vi nghiên cứu.

Đối tượng nghiên cứu: ML và thuật toán K-Nearest Neighbors.

Phạm vi nghiên cứu của đề tài tập trung vào thuật toán K-Nearest Neighbors.

Phương pháp nghiên cứu.

Chủ động tiến hành thu thập và phân tích các tài liệu, thông tin liên quan đến đề tài. Từ đó, lựa chọn phương hướng giải quyết vấn đề, tìm hiểu thuật toán và ứng dụng của thuật toán.

Ý nghĩa khoa học.

Việc xây dựng được các hệ thống dự báo, dự đoán tốt giúp ích rất nhiều cho các cá nhân, tổ chức, doanh nghiệp.

- Đối với các cá nhân, những dự báo chính xác sẽ giúp cải thiện đời sống vật chất cũng như tinh thần, giúp con người có cái nhìn đa chiều hơn về sự vật, hiện tượng đó.
- Đối với các doanh nghiệp, nếu công tác dự báo được thực hiện một cách nghiêm túc, chính xác sẽ tạo điều kiện nâng cao khả năng cạnh tranh trên thị trường, đồng thời giảm thiểu những rủi ro, tổn thất không mong muốn.
- Đối với các tổ chức xã hội, những dự báo chính xác sẽ là căn cứ để các nhà lãnh đạo có thể đưa ra các chính sách phát triển kinh tế, văn hoá, xã hội phù hợp nhằm đưa nền kinh tế ngày càng phát triển.

Học máy là một phần không thể thiếu trong nhiều dự án nghiên cứu và ứng dụng thương mại ngày nay, trong các lĩnh vực từ chẩn đoán và điều trị y tế đến tìm kiếm bạn bè của bạn trên mạng xã hội. Nhiều người nghĩ rằng học máy chỉ có thể được áp dụng bởi các công ty lớn với đội ngũ nghiên cứu sâu rộng. Trong cuốn báo cáo này, tôi muốn cho thấy việc tự xây dựng các giải pháp học máy dễ dàng như thế nào và cách thực hiện nó một cách hiệu quả nhất. Các ứng dụng của máy học là vô tận và với số lượng dữ liệu có sẵn ngày nay, hầu hết bị giới hạn bởi trí tưởng tượng của mình.

Cấu trúc nội dung báo cáo.

Đồ án được chia thành 4 chương như sau:

- Chương 1: Cơ sở lý thuyết.
- Chương 2: Phân tích – Đánh giá.
- Chương 3: Ứng dụng thuật toán K-Nearest Neighbors vào bài toán dự đoán tình trạng hôn nhân.
- Chương 4: Kết luận và kiến nghị.

Kết luận tổng quan.

Trong phần này, tôi đã giới thiệu khái quát tổng quan những vấn đề mà đề tài nghiên cứu.

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

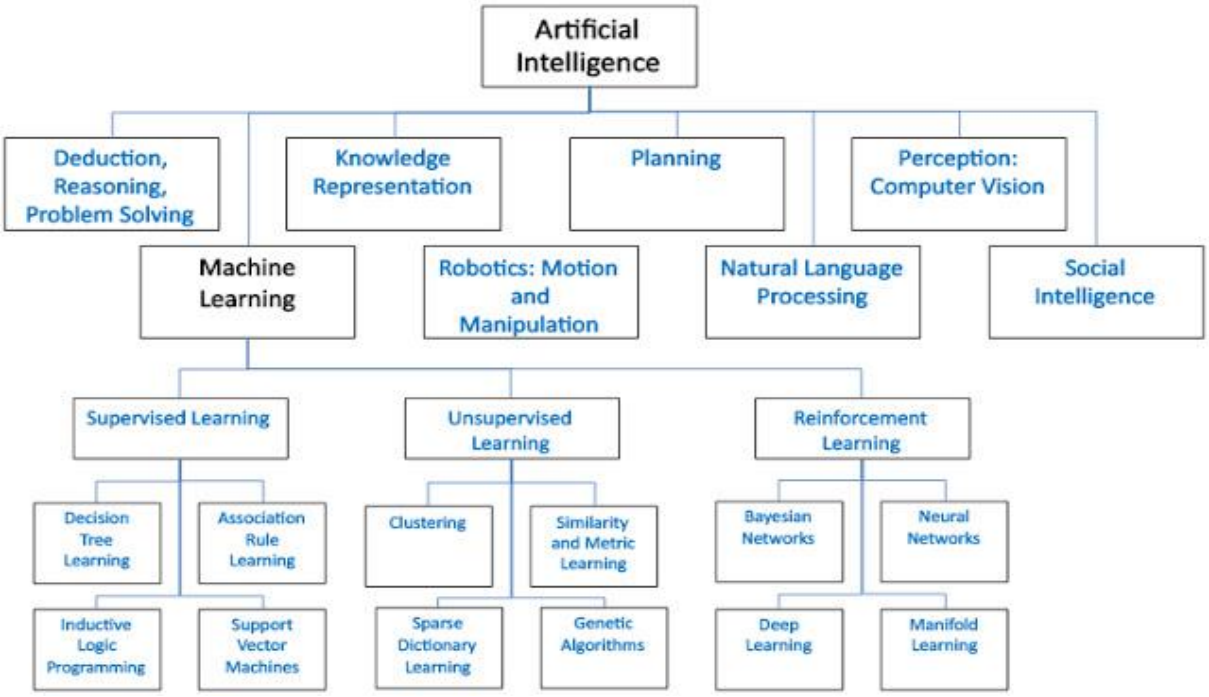
1.1. Machine Learning.

1.1.1. Giới thiệu về Machine Learning.

Một sự thật hiển nhiên, ngày nay mọi người hầu như đều nói về AI: AI đang thay đổi thế giới, AI là một yếu tố cần thiết cho sự cải tiến, AI được ứng dụng vào rất rất nhiều thứ,…Ngay cả các trường Đại học trên thế giới đều muốn thêm khóa học AI vào chương trình giảng dạy của trường.

Một định nghĩa chính thức về AI trong Khoa học máy tính là: việc nghiên cứu và thiết kế bất kỳ tác nhân thông minh nào nhận thức được môi trường của nó và thực hiện các hành động để tối đa hóa cơ hội đạt được thành công các mục tiêu của nó.

Và Machine Learning chính là một tập con trong trí tuệ nhân tạo. Nó là một lĩnh vực nhỏ trong ngành khoa học máy tính, có khả năng tự học hỏi dựa trên dữ liệu được đưa vào mà không cần phải lập trình cụ thể.



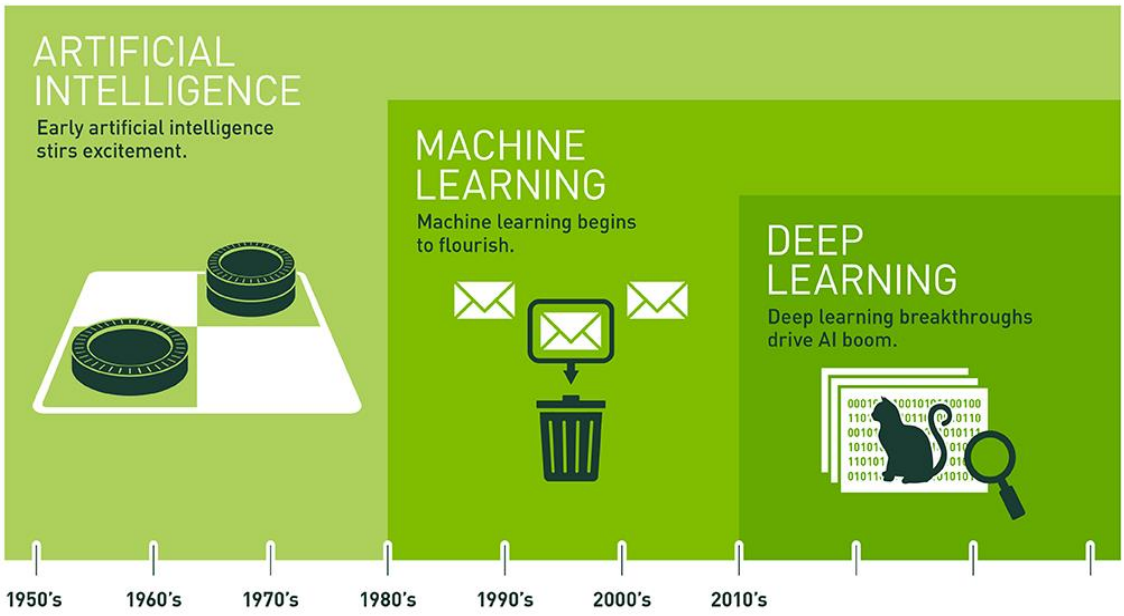
Hình 1. 1 Các lĩnh vực trong Trí tuệ nhân tạo

Có thể thấy, AI bao gồm 8 lĩnh vực lớn là:

- Suy luận, Lập luận, Giải quyết vấn đề (Deduction, Reasoning, Problem Solving).
- Biểu diễn tri thức (Knowledge Representaion).
- Hoạch định (Planning).
- Nhận thức: Thị giác máy tính (Perception: Computer Vision).

- Máy học/Học máy (Machine Learning).
- Người máy: Chuyển động và Thao tác (Robotics: Motion and Manipulation).
- Xử lý ngôn ngữ tự nhiên (Natural Language Processing).
- Trí thông minh xã hội (Social Intelligence).

AI là một lĩnh vực nghiên cứu rất, rất rộng lớn và có tin hay không, chưa bao giờ có một phân loại chính thức được chấp nhận cho các nhánh của AI.



Hình 1. 2 Mối quan hệ giữa AI, Machine Learning và Deep Learning

Trong đó, Deep Learning là 1 nhánh nhỏ trong lĩnh vực Học củng cố (Reinforcement Learning) của Machine Learning, còn Machine Learning là 1 phần của AI. Đề xuất AI được đưa ra vào năm 1943, sau đó tiến độ đã bị chậm lại và gần như dừng lại vào năm 1974, "mùa đông AI". Đến đầu năm 1980, nghiên cứu AI đã được hồi sinh nhờ hệ chuyên gia thành công thương mại của các hệ chuyên gia (ML mô phỏng kiến thức và kỹ năng phân tích của các chuyên gia về con người). Vào khoảng năm 2012 với lượng dữ liệu khổng lồ (Big Data), máy tính nhanh hơn, cải tiến thuật toán và truy cập vào lượng lớn dữ liệu cho phép có được các tiến bộ trong học tập và nhận thức máy; phương pháp Học sâu (Deep Learning) vốn đói dữ liệu bắt đầu thống trị các thử nghiệm liên quan đến độ chính xác. Đây là tổng quan về mối quan hệ và lịch sử hình thành của AI, ML, DL.

Như cách nói của Andreas C. Müller và Sarah Guido trong tài liệu tham khảo số [3], Máy học là cách trích xuất kiến thức từ dữ liệu. Nó là một lĩnh vực nghiên cứu ở

giao điểm của thống kê, trí tuệ nhân tạo và khoa học máy tính và còn được gọi là phân tích dự đoán hoặc học thống kê. Việc áp dụng các phương pháp học máy trong những năm gần đây đã trở nên phổ biến trong cuộc sống hàng ngày. Từ các đề xuất tự động về bộ phim nên xem, món ăn nên đặt hoặc sản phẩm cần mua, đài phát thanh trực tuyến được cá nhân hóa và nhận dạng bạn bè trong ảnh của bạn, nhiều trang web và thiết bị hiện đại có các thuật toán máy học ở cốt lõi của chúng. Khi bạn xem một trang web phức tạp như Facebook, Amazon hoặc Netflix, rất có thể mọi phần của trang web đều chứa nhiều mô hình học máy.

Ngoài các ứng dụng thương mại, học máy đã có ảnh hưởng to lớn đến cách thức thực hiện nghiên cứu theo hướng dữ liệu ngày nay. Trong chương này, tôi sẽ định nghĩa về ML cũng như giải thích tại sao máy học lại trở nên phổ biến và thảo luận về những loại vấn đề nào có thể được giải quyết bằng cách sử dụng máy học.

a) Định nghĩa Machine Learning.

Tác giả cuốn sách "Understanding Machine Learning: From Theory to Algorithms" [1] đã đề cập, ML là làm cho máy có thể học tự động. Đó là lập trình máy tính để chúng có thể "học" từ đầu vào có sẵn cho chúng. Nói một cách đại khái, học là quá trình chuyển đổi kinh nghiệm thành chuyên môn hoặc kiến thức. Đầu vào cho một thuật toán học tập là dữ liệu đào tạo, đại diện cho kinh nghiệm và đầu ra là một số kiến thức chuyên môn, thường có dạng một chương trình máy tính có thể thực hiện một số nhiệm vụ khác.

Machine Learning là một thuật toán có khả năng học tập từ dữ liệu, có nghĩa là chương trình máy tính sẽ học hỏi từ kinh nghiệm E (Experience) từ các tác vụ T (Task), với kết quả được đo bằng hiệu suất P (Performance). Nếu hiệu suất của nó áp dụng trên tác vụ T khi được đánh giá bởi P và cải thiện theo kinh nghiệm E.

Ví dụ 1: Giả sử như bạn muốn máy tính xác định một tin nhắn có phải là SPAM hay không?

- Tác vụ T: Xác định 1 tin nhắn có phải SPAM hay không?
- Kinh nghiệm E: Xem lại những tin nhắn đánh dấu là SPAM xem có những đặc tính gì để có thể xác định nó là SPAM.
- Độ đo P: Là phần trăm số tin nhắn SPAM được phân loại đúng.

Ví dụ 2: Chương trình nhận dạng số (số từ 0 -> 9)

- T: Là nhận dạng được ảnh chứa ký tự số.

- E: Đặc trưng để phân loại ký tự số từ tập dữ liệu số cho trước.
- P: Độ chính xác của quá trình nhận dạng.

b) Lý do và khi nào sử dụng Machine Learning.

Trong những ngày đầu của các ứng dụng "thông minh", nhiều hệ thống đã sử dụng các quy tắc được mã hóa thủ công về các quyết định "nếu" và "ngược lại" để xử lý dữ liệu hoặc điều chỉnh theo đầu vào của người dùng. Hãy nghĩ đến một bộ lọc thư rác có công việc là chuyển các thư email đến thích hợp vào một thư mục thư rác. Bạn có thể tạo ra một danh sách đen các từ dẫn đến một email bị đánh dấu là spam. Đây sẽ là một ví dụ về việc sử dụng hệ thống quy tắc do chuyên gia thiết kế để thiết kế một ứng dụng "thông minh". Việc xây dựng các quy tắc quyết định theo cách thủ công là khả thi đối với một số ứng dụng, đặc biệt là những ứng dụng trong đó con người hiểu rõ về quy trình để lập mô hình. Tuy nhiên, việc sử dụng các quy tắc được mã hóa bằng tay để đưa ra quyết định có hai nhược điểm lớn:

- Logic cần thiết để đưa ra quyết định là cụ thể cho một miền và nhiệm vụ. Thay đổi nhiệm vụ thậm chí một chút có thể yêu cầu viết lại toàn bộ hệ thống.
- Việc thiết kế các quy tắc đòi hỏi sự hiểu biết sâu sắc về cách một người chuyên gia nên đưa ra quyết định.

Một ví dụ về trường hợp phương pháp mã hóa tay này sẽ thất bại là phát hiện khuôn mặt trong hình ảnh. Ngày nay, mọi điện thoại thông minh đều có thể phát hiện khuôn mặt trong ảnh. Tuy nhiên, nhận diện khuôn mặt vẫn là một vấn đề chưa được giải quyết cho đến tận năm 2001. Vấn đề chính là cách mà các pixel (tạo nên hình ảnh trong máy tính) được máy tính "cảm nhận" rất khác với cách con người nhận biết khuôn mặt. Sự khác biệt về cách thể hiện này khiến con người về cơ bản không thể đưa ra một bộ quy tắc tốt để mô tả những gì tạo nên một khuôn mặt trong một hình ảnh kỹ thuật số.

Tuy nhiên, sử dụng máy học, chỉ cần trình bày một chương trình với một bộ sưu tập lớn các hình ảnh về khuôn mặt là đủ để thuật toán xác định những đặc điểm nào cần thiết để nhận dạng khuôn mặt.

Để biết khi nào cần học máy thay vì lập trình trực tiếp máy tính của mình để thực hiện nhiệm vụ trong tầm tay, ta xét hai phương diện của vấn để được đưa ra là các chương trình có cần học hỏi và cải thiện dựa trên "kinh nghiệm" của chúng. Cụ thể là sự phức tạp của vấn đề và nhu cầu thích ứng:

- Các nhiệm vụ quá phức tạp để lập trình:
  - Nhiệm vụ do Động vật/Con người thực hiện: Ví dụ lái xe, nhận dạng và hiểu hình ảnh. Trong tất cả các nhiệm vụ này, các chương trình "học từ kinh nghiệm của chúng" (đạt được kết quả khá khả quan sau khi tiếp xúc để có đủ nhiều dữ liệu đào tạo).
  - Nhiệm vụ vượt quá khả năng của con người: liên quan đến phân tích của các bộ dữ liệu rất lớn và phức tạp: dữ liệu thiên văn, biến dữ liệu lưu trữ y tế thành kiến thức y học, dự đoán thời tiết, phân tích của dữ liệu bộ gen, công cụ tìm kiếm web và thương mại điện tử.
- Tính thích nghi:

  Một tính năng hạn chế của các công cụ được lập trình là độ cứng của chúng – khi chương trình đã được viết ra và cài đặt, nó vẫn không thay đổi. Tuy nhiên, nhiều tác vụ thay đổi theo thời gian hoặc từ người dùng này sang người dùng khác. Các công cụ học máy - các chương trình có hành vi thích ứng với dữ liệu đầu vào của chúng - đưa ra giải pháp cho những vấn đề như vậy; về bản chất, chúng thích nghi với những thay đổi trong môi trường mà chúng tương tác. Các ứng dụng thành công điển hình của máy học cho những vấn đề như vậy bao gồm các chương trình giải mã văn bản viết tay, trong đó một chương trình cố định thích ứng với các biến thể giữa chữ viết tay của những người dùng khác nhau; chương trình phát hiện thư rác, một chương trình tự động thích ứng với những thay đổi về bản chất của thư rác; và các chương trình nhận dạng giọng nói.

c) Đối tượng sử dụng

Hầu hết mọi ngành công nghiệp đang làm việc với hàm lượng lớn dữ liệu đều nhận ra tầm quan trọng của công nghệ ML. Những cái nhìn sâu sắc từ nguồn dữ liệu này, sẽ giúp các tổ chức vận hành hiệu quả hơn hoặc tạo được lợi thế cạnh tranh so với các đối thủ.

- Các dịch vụ tài chính

  Ngân hàng và những doanh nghiệp hoạt động trong lĩnh vực tài chính sử dụng công nghệ ML với 2 mục đích chính: xác định insights trong dữ liệu và ngăn chặn lừa đảo. Insights sẽ biết được các cơ hội đầu tư hoặc thông báo đến nhà đầu tư thời điểm giao dịch hợp lý. Khai phá dữ liệu cũng có thể

tìm được những khách hàng đang có hồ sơ rủi ro cao hoặc sử dụng giám sát mạng để chỉ rõ những tín hiệu lừa đảo.

- Chính phủ

  Các tổ chức chính phủ hoạt động về an ninh cộng đồng hoặc tiện ích xã hội sở hữu rất nhiều nguồn dữ liệu có thể khai thác insights. Ví dụ, khi phân tích dữ liệu cảm biến, chính phủ sẽ tăng mức độ hiệu quả của dịch vụ và tiết kiệm chi phí. ML còn hỗ trợ phát hiện gian lận và giảm thiểu khả năng trộm cắp danh tính.

- Chăm sóc sức khỏe

  ML là 1 xu hướng phát triển nhanh chóng trong ngành chăm sóc sức khỏe, nhờ vào sự ra đời của các thiết bị và máy cảm ứng đeo được sử dụng dữ liệu để đánh giá tình hình sức khỏe của bệnh nhân trong thời gian thực. Công nghệ ML còn giúp các chuyên gia y tế xác định những xu hướng hoặc tín hiệu để cải thiện khả năng điều trị, chẩn đoán bệnh.

- Marketing và sales

  Dựa trên hành vi mua hàng trước đây, các trang website sử dụng ML phân tích lịch sử mua hàng, từ đó giới thiệu những vật dụng mà bạn có thể sẽ quan tâm và yêu thích. Khả năng tiếp nhận dữ liệu, phân tích và sử dụng những dữ liệu đó để cá nhân hóa trải nghiệm mua sắm hoặc thực hiện chiến dịch Marketing chính là tương tai của ngành bán lẻ.

- Dầu khí

  Tìm kiếm những nguồn nguyên liệu mới. Phân tích các mỏ dầu dưới đất. Dự đoán tình trạng thất bại của bộ cảm biến lọc dầu. Sắp xếp các kênh phân phối để đạt hiệu quả và tiết kiệm chi phí. Có thể nói, số lượng các trường hợp sử dụng ML trong ngành công nghiệp này cực kì lớn và vẫn ngày càng mở rộng.

- Vận tải

  Phân tích dữ liệu để xác định mô hình và các xu hướng là trọng tâm trong ngành vận tải vì đây là ngành phụ thuộc vào khả năng tận dụng hiệu quả trên mỗi tuyến đường và dự đoán các vấn đề tiềm tàng để gia tăng lợi nhuận. Các chức năng phân tích dữ liệu và modeling của ML đóng vai trò quan

trọng với các doanh nghiệp vận chuyển, vận tải công cộng và các tổ chức vận chuyển khác.

1.1.2. Phân nhóm các thuật toán Machine Learning.

Các thuật toán ML thường được chia làm 4 nhóm:

- Học có giám sát (Supervise learning).
- Học không giám sát (Unsupervised learning).
- Học bán giám sát (Semi-supervised lerning).
- Học củng cố (Reinforcement learning).

a. Supervised Learning (Học có giám sát).

Các thuật toán học máy học từ các cặp đầu vào / đầu ra được gọi là các thuật toán học có giám sát (SL) vì cung cấp sự giám sát đối với các thuật toán trong dạng đầu ra mong muốn cho từng ví dụ mà chúng học được. Trong khi tạo một tập dữ liệu đầu vào và đầu ra thường là một quá trình thủ công tốn nhiều công sức, các thuật toán học tập có giám sát được hiểu rõ và hiệu suất của chúng rất dễ đo lường. Nếu là ứng dụng có thể được xây dựng như một vấn đề học tập có giám sát và có thể tạo tập dữ liệu bao gồm kết quả mong muốn, máy học sẽ có thể giải quyết vấn đề. Supervised learning là nhóm phổ biến nhất trong các thuật toán Machine Learning.

Một cách toán học, Supervised learning là khi chúng ra có một tập hợp biến đầu vào $X=\{x1,x2,\ldots,xN\}$ và một tập hợp nhãn tương ứng $Y=\{y1,y2,\ldots,yN\}$ trong đó xi,yi là các vector. Các cặp dữ liệu biết trước $(xi,yi)\in X\times Y$ được gọi là tập training data (dữ liệu huấn luyện). Từ tập traing data này, chúng ta cần tạo ra một hàm số ánh xạ mỗi phần tử từ tập X sang một phần tử (xấp xỉ) tương ứng của tập Y: $yi\approx f(xi),\ \forall i=1,2,\ldots,N$ Mục đích là xấp xỉ hàm số f thật tốt để khi có một dữ liệu x mới, chúng ta có thể tính được nhãn tương ứng của nó y=f(x).

Các ví dụ điển hình: Nhận dạng mã zip từ các chữ số viết tay trên phong bì, Xác định xem khối u có lành tính hay không dựa trên hình ảnh y tế, Phát hiện hoạt động lừa đảo trong giao dịch thẻ tín dụng,…

Một điều thú vị cần lưu ý về những ví dụ này là mặc dù đầu vào và đầu ra trông khá đơn giản, nhưng quá trình thu thập dữ liệu cho ba tác vụ này rất khác nhau. Trong khi đọc phong bì là công sức, nó dễ dàng và rẻ. Mặt khác, để có được các chẩn đoán và hình ảnh y tế, không chỉ đòi hỏi máy móc đắt tiền mà còn phải có kiến thức chuyên môn thật vững, chưa kể đến các vấn đề về đạo đức và quyền riêng tư. Trong ví dụ phát hiện

gian lận thẻ tín dụng, việc thu thập dữ liệu đơn giản hơn nhiều. Khách hàng sẽ cung cấp đầu ra mong muốn, cũng như họ sẽ báo cáo lừa đảo. Tất cả những gì phải làm để có được các cặp đầu vào / đầu ra của hoạt động lừa đảo và không lừa đảo là chờ đợi.

Ví dụ: trong nhận dạng chữ viết tay, ta có ảnh của hàng nghìn ví dụ của mỗi chữ số được viết bởi nhiều người khác nhau. Chúng ta đưa các bức ảnh này vào trong một thuật toán và chỉ cho nó biết mỗi bức ảnh tương ứng với chữ số nào. Sau khi thuật toán tạo ra một mô hình, tức một hàm số mà đầu vào là một bức ảnh và đầu ra là một chữ số, khi nhận được một bức ảnh mới mà mô hình chưa nhìn thấy bao giờ, nó sẽ dự đoán bức ảnh đó chứa chữ số nào.

Ví dụ này khá giống với cách học của con người khi còn nhỏ. Ta đưa bảng chữ cái cho một đứa trẻ và chỉ cho chúng đây là chữ A, đây là chữ B. Sau một vài lần được dạy thì trẻ có thể nhận biết được đâu là chữ A, đâu là chữ B trong một cuốn sách mà chúng chưa nhìn thấy bao giờ.

Thuật toán supervised learning còn được tiếp tục chia nhỏ ra thành hai loại chính:

Classification (Phân loại)

Một bài toán được gọi là classification nếu các label của input data được chia thành một số hữu hạn nhóm.

Ví dụ: Gmail xác định xem một email có phải là spam hay không; các hãng tín dụng xác định xem một khách hàng có khả năng thanh toán nợ hay không. Ba ví dụ phía trên được chia vào loại này.

Regression (Hồi quy)

Nếu label không được chia thành các nhóm mà là một giá trị thực cụ thể.

Ví dụ: một căn nhà rộng x m2, có y phòng ngủ và cách trung tâm thành phố z km sẽ có giá là bao nhiêu?

b. Unsupervised Learning (Học không giám sát).

Trong thuật toán này, chúng ta không biết được outcome hay nhãn mà chỉ có dữ liệu đầu vào. Thuật toán Unsupervised learning sẽ dựa vào cấu trúc của dữ liệu để thực hiện một công việc nào đó, ví dụ như phân nhóm hoặc giảm số chiều của dữ liệu để thuận tiện trong việc lưu trữ và tính toán.

Một cách toán học, Unsupervised learning là khi chúng ta chỉ có dữ liệu vào X mà không biết nhãn Y tương ứng.

Những thuật toán loại này được gọi là Unsupervised learning vì không giống như Supervised learning, chúng ta không biết câu trả lời chính xác cho mỗi dữ liệu đầu vào. Giống như khi ta học, không có thầy cô giáo nào chỉ cho ta biết đó là chữ A hay chữ B. Cụm không giám sát được đặt tên theo nghĩa này. Các ứng dụng của thuật toán này thường khó hiểu và khó đánh giá hơn.

Ví dụ về học tập không có giám sát bao gồm:

- Xác định chủ đề trong một tập hợp các bài đăng trên blog: Nếu có một bộ sưu tập lớn dữ liệu văn bản, bạn có thể muốn tóm tắt nó và tìm các chủ đề thịnh hành trong đó. Bạn có thể không biết trước những chủ đề này là gì hoặc có thể có bao nhiêu chủ đề. Do đó, không có đầu ra nào được biết đến.
- Phân khúc khách hàng thành các nhóm có sở thích giống nhau: Với một bộ hồ sơ khách hàng, bạn có thể muốn xác định khách hàng nào là tương tự và liệu có những nhóm khách hàng có sở thích giống nhau hay không. Đối với trang web mua sắm, đây có thể là “cha mẹ”, “mọt sách” hoặc “game thủ”. Bởi vì không biết trước những nhóm này có thể là gì, hoặc thậm chí có bao nhiêu nhóm, nên không có đầu ra nào được biết đến.
- Phát hiện các kiểu truy cập bất thường vào một trang web: Để xác định lạm dụng hoặc lỗi, thường hữu ích khi tìm các mẫu truy cập khác với tiêu chuẩn. Mỗi kiểu bất thường có thể rất khác nhau và có thể không có bất kỳ trường hợp hành vi bất thường nào được ghi lại. Bởi vì trong ví dụ này, chỉ quan sát lưu lượng truy cập và không biết điều gì tạo nên hành vi bình thường và bất thường, đây là một vấn đề không được giám sát.

Các bài toán Unsupervised learning được tiếp tục chia nhỏ thành hai loại:

Clustering (phân nhóm)

Một bài toán phân nhóm toàn bộ dữ liệu X thành các nhóm nhỏ dựa trên sự liên quan giữa các dữ liệu trong mỗi nhóm.

Ví dụ: phân nhóm khách hàng dựa trên hành vi mua hàng. Điều này cũng giống như việc ta đưa cho một đứa trẻ rất nhiều mảnh ghép với các hình thù và màu sắc khác nhau, ví dụ tam giác, vuông, tròn với màu xanh và đỏ, sau đó yêu cầu trẻ phân chúng thành từng nhóm. Mặc dù không cho trẻ biết mảnh nào tương ứng với hình nào hoặc màu nào, nhiều khả năng chúng vẫn có thể phân loại các mảnh ghép theo màu hoặc hình dạng.

Association (Suy diễn)

Là bài toán khi chúng ta muốn khám phá ra một quy luật dựa trên nhiều dữ liệu cho trước.

Ví dụ: những khách hàng nam mua quần áo thường có xu hướng mua thêm đồng hồ hoặc thắt lưng; những khán giả xem phim Spider Man thường có xu hướng xem thêm phim Bat Man, dựa vào đó tạo ra một hệ thống gợi ý khách hàng, thúc đẩy nhu cầu mua sắm.

c. Semi-Supervised Learning (Học bán giám sát).

Các bài toán khi chúng ta có một lượng lớn dữ liệu X nhưng chỉ một phần trong chúng được gán nhãn được gọi là Semi-Supervised Learning. Những bài toán thuộc nhóm này nằm giữa hai nhóm được nêu bên trên.

Một ví dụ điển hình của nhóm này là chỉ có một phần ảnh hoặc văn bản được gán nhãn (ví dụ bức ảnh về người, động vật hoặc các văn bản khoa học, chính trị) và phần lớn các bức ảnh, văn bản khác chưa được gán nhãn được thu thập từ internet. Thực tế cho thấy rất nhiều các bài toán ML thuộc vào nhóm này vì việc thu thập dữ liệu có nhãn tốn rất nhiều thời gian và có chi phí cao. Rất nhiều loại dữ liệu thậm chí cần phải có chuyên gia mới gán nhãn được. Ngược lại, dữ liệu chưa có nhãn có thể được thu thập với chi phí thấp từ internet.

d. Reinforcement Learning (Học Củng Cố)

Reinforcement learning là các bài toán giúp cho một hệ thống tự động xác định hành vi dựa trên hoàn cảnh để đạt được lợi ích cao. Hiện tại, Reinforcement Learning chủ yếu được áp dụng vào Lý Thuyết Trò Chơi, các thuật toán cần xác định nước đi tiếp theo để đạt được điểm số cao nhất.

Ví dụ: AlphaGo gần đây nổi tiếng với việc chơi cờ vây thắng cả con người. Cờ vây được xem là có độ phức tạp cực kỳ cao với tổng số nước đi là xấp xỉ 1076110761, so với cờ vua là 1012010120 và tổng số nguyên tử trong toàn vũ trụ là khoảng 10801080.

Vì vậy, thuật toán phải chọn ra 1 nước đi tối ưu trong số hàng nhiều tỉ tỉ lựa chọn, và tất nhiên, không thể áp dụng thuật toán tương tự như IBM Deep.

Về cơ bản, AlphaGo bao gồm các thuật toán thuộc cả Supervised learning và Reinforcement Learning. Trong phần Supervised Learning, dữ liệu từ các ván cờ do con

người chơi với nhau được đưa vào để huấn luyện. Tuy nhiên, mục đích cuối cùng của AlphaGo không phải là chơi như con người mà phải thậm chí thắng cả con người.

Vì vậy, sau khi học xong các ván cờ của con người, AlphaGo tự chơi với chính nó với hàng triệu ván chơi để tìm ra các nước đi mới tối ưu hơn. Thuật toán trong phần tự chơi này được xếp vào loại Reinforcement learning.

1.1.3. Các bước thực hiện Machine Learning.

Thực hiện Machine Learning bao gồm các bước như sau:

Thu thập và chuẩn bị dữ liệu:

Yếu tố ban đầu cần thiết để thực hiện ML là cần có dữ liệu. Dữ liệu có thể được lấy từ nhiều nguồn khác nhau, có thể ít, có thể nhiều, có thể sạch, có thể nhiều dữ liệu lỗi…Sau khi thu thập, dữ liệu cần được làm sạch.

Chọn thành phần

Với mỗi tập dữ liệu có thể có rất nhiều thành phần, nhưng không phải thành phần nào cũng liên quan tới bài toán mà ta cần giải, việc lựa chọn thành phần (loại bỏ các thành phần không cần thiết) giúp cho việc học của ta trở nên nhanh và hiệu quả hơn. Tuy nhiên, việc lựa chọn đòi hỏi sự thấu hiểu về dữ liệu và bài toán, chủ yếu làm bằng tay và sức người.

Sau khi chọn được thành phần, nhiều khi ta quay lại bước 1, tiến hành loại bỏ các dữ liệu không liên quan để thu nhỏ tập dữ liệu.

Việc thu nhỏ tập dữ liệu cũng khiến cho việc học của ta tốn ít thời gian hơn, tuy nhiên dữ liệu ít quá cũng khiến việc học đạt độ chính xác không cao, cần cân đối giữa các yếu tố này.

Chuẩn hoá dữ liệu

Nhiều khi dữ liệu của từng thành phần có định dạng, kích thước khác biệt lớn, ví dụ thành phần 1 có dữ liệu trong khoảng [0, 1], thành phần 2 có dữ liệu trong khoảng [-1000, 1000], thành phần 3 có dữ liệu là hình ảnh… Để tăng tốc độ và hiệu quả của việc học, ta cần chuẩn hoá dữ liệu, đưa dữ liệu của tất cả các thành phần về cùng một định dạng (số hoá), và cùng khoảng biến thiên.

Chọn thuật toán

Tuỳ vào dữ liệu, bài toán mà ta lựa chọn thuật toán ML tương ứng. Đôi khi lựa chọn thuật toán cũng cần dựa trên kinh nghiệm. Một số gợi ý cho việc lựa chọn thuật toán là tham khảo của Scikit-learn.

Chọn parameter cho thuật toán

Tuỳ mỗi thuật toán mà có nhiều các cách cài đặt khác nhau. Hơn thế nữa các tham số tính toán cũng quyết định không nhỏ tới kết quả tính toán. Vì thế khi sau khi chọn thuật toán thì việc chọn parameter phù hợp với dữ liệu cũng khá quan trọng. Việc chọn Parameter chủ yếu dựa trên kinh nghiệm, nhiều khi có thể sử dụng các thư viện support.

Huấn luyện và Đánh giá

Sau khi chọn được thuật toán (một hoặc nhiều thuật toán) và parameter tương ứng ta cho dữ liệu vào train. Tiến hành cross-validation để điều chỉnh model.

Sau khi train được model, ta đưa data test vào kiểm tra và đánh giá độ chính xác của model vừa train được.

Phân chia dữ liệu

Thông thường với tập data cho trước ta thường chia làm 3 tập để sử dụng với mục đích khác nhau:

- **Tập huấn luyện** (Training set): dùng để huấn luyện model.
- **Tập kiểm chứng** (Validation set): dùng để đánh giá, điều chỉnh model. Ví dụ như ta dùng tập huấn luyện cho nhiều thuật toán khác nhau rồi dùng tập kiểm chứng để chọn thuật toán phù hợp nhất. Hoặc tìm parameter phù hợp nhất cho một thuật toán cụ thể.
- **Tập kiểm tra** (Test set): dùng để đánh giá model sau khi huấn luyện. Mô hình đã được đánh giá bằng tập test phải là mô hình cuối cùng, không được thay đổi nữa.

Nếu model cuối cùng thu được sau khi huấn luyện và kiểm chứng cần phải được đánh giá bằng tập kiểm tra. Nếu kết quả không tốt, cần thực hiện huấn luyện lại từ đầu.

1.1.4. Các ứng dụng.

Có rất nhiều ứng dụng mang tính thực tế cao của máy học mà khó có thể kể hết được. Những ứng dụng dưới đây là những ứng dụng phổ biến và được chọn lọc theo góc nhìn cá nhân.

- Nhận diện và phát hiện khuôn mặt: Nhận diện và phát hiện khuôn mặt là ứng dụng khá thú vị của máy học và được áp dụng khá nhiều vào đời sống.

Tiêu biểu là tính năng phát hiện khuôn mặt ở máy chụp ảnh. Ứng dụng được phát triển thêm thành phát hiện chớp mắt, phát hiện cười....

- Xe tự lái: Xe tự lái mặc dù phát triển từ đầu thập niên 90 nhưng cho tới nay vẫn còn là vấn đề được nhiều người quan tâm. Các hãng lớn như Google, NVDIA đang nỗ lực để tạo ra một cỗ máy có thể hoàn toàn tự động lái xe và giảm thiểu tai nạn cho con người.
- Phân lớp ảnh: Tìm kiếm ảnh trên Google hiện rất quen thuộc với nhiều người. Ứng dụng của phân lớp ảnh giúp người dùng sử dụng ảnh làm từ khóa tìm kiếm thay thế cho việc tìm kiếm truyền thống trên Google. Bạn upload một ảnh lên và Google sẽ giúp bạn tìm kiếm những thông tin liên quan đến bức ảnh đó.
- Nhận dạng giọng nói: Các trợ lý ảo như Siri, Conrtana hay Google Now là ví dụ điển hình cho nhận dạng giọng nói. Một ví dụ khác nữa là tính năng dịch thuật trực tuyến của Youtube. Với ứng dụng của DL, khả năng dịch thuật chính xác ngôn ngữ từ các video Youtube đang ngày một phát triển vượt bậc.
- Anti-virus: Có thể nhiều người không nghĩ rằng các phần mềm diệt virus lại áp dụng máy học. Tuy nhiên, áp dụng máy học vào để phân tích và dự đoán xu hướng các loại virus sẽ giúp ích rất nhiều trong việc bảo vệ dữ liệu máy tính.

1.2. Python.

1.2.1. Giới thiệu về Python.

Python là một ví dụ về ngôn ngữ cấp cao; các ngôn ngữ cấp cao khác có thể đã nghe nói đến là C, C ++, PHP, Ruby, Basic, Perl, Java và JavaScript (có thể tham khảo trong tài liệu [5]).

Ngoài ra còn có các ngôn ngữ cấp thấp, đôi khi được gọi là “ngôn ngữ máy” hoặc “hợp ngữ”. Nói một cách dễ hiểu, máy tính chỉ có thể chạy các chương trình được viết bằng các ngôn ngữ cấp thấp. Vì vậy, các chương trình được viết bằng ngôn ngữ cấp cao phải được xử lý trước khi chúng có thể chạy. Quá trình xử lý bổ sung này mất một thời gian, đây là một nhược điểm nhỏ của các ngôn ngữ cấp cao.

Nhưng lợi thế là rất lớn. Đầu tiên, việc lập trình bằng ngôn ngữ cấp cao sẽ dễ dàng hơn nhiều. Các chương trình được viết bằng ngôn ngữ cấp cao tốn ít thời gian hơn

để viết, chúng ngắn hơn và dễ đọc hơn, và chúng có nhiều khả năng chính xác hơn. Thứ hai, các ngôn ngữ cấp cao có tính di động, có nghĩa là chúng có thể chạy trên các loại máy tính khác nhau với ít hoặc không có sửa đổi. Các chương trình cấp thấp chỉ có thể chạy trên một loại máy tính và phải được viết lại để chạy trên một loại máy tính khác.

Nó là một ngôn ngữ lập trình có mục đích chung được sử dụng cho mọi thứ từ phát triển web (web-development) đến học sâu (Deep Learning). Theo một số số liệu, nó được xếp hạng là một trong ba ngôn ngữ phổ biến nhất. Hiện nó là ngôn ngữ giới thiệu được dạy thường xuyên nhất tại các trường đại học hàng đầu của Hoa Kỳ theo một bài báo trên blog ACM gần đây. Do tính phổ biến của nó, Python có một cộng đồng nguồn mở phát triển mạnh và có hơn 80.000 gói phần mềm miễn phí có sẵn cho ngôn ngữ trên Chỉ mục gói Python chính thức (PyPI)

Cú pháp của Python là khá dễ dàng để học và ngôn ngữ này cũng mạnh mẽ và linh hoạt không kém các ngôn ngữ khác trong việc phát triển các ứng dụng. Python hỗ trợ mẫu đa lập trình, bao gồm lập trình hướng đối tượng, lập trình hàm và mệnh lệnh hoặc là các phong cách lập trình theo thủ tục.

Python không chỉ làm việc trên lĩnh vực đặc biệt như lập trình web, và đó là tại sao ngôn ngữ này là đa mục đích bởi vì nó có thể được sử dụng với web, enterprise, 3D CAD, …

1.2.2. Đặc điểm của Python.

Dưới đây là một số đặc điểm chính của Python:

- Dễ dàng để sử dụng: Python là một ngôn ngữ bậc cao rất dễ dàng để sử dụng. Python có một số lượng từ khóa ít hơn, cấu trúc của Python đơn giản hơn và cú pháp của Python được định nghĩa khá rõ ràng, … Tất cả các điều này làm Python thực sự trở thành một ngôn ngữ thân thiện với lập trình viên.
- Bạn có thể đọc code của Python khá dễ dàng. Phần code của Python được định nghĩa khá rõ ràng và rành mạch.
- Python có một thư viện chuẩn khá rộng lớn. Thư viện này dễ dàng tương thích và tích hợp với UNIX, Windows, và Macintosh.
- Python cho phép chia nhỏ chương trình của mình ra thành các mô-đun để có thể sử dụng lại trong các chương trình Python khác. Nó có sẵn rất nhiều các mô-đun chuẩn để có thể sử dụng làm cơ sở cho chương trình -- hoặc

như các ví dụ để bắt đầu học lập trình bằng Python. Một vài mô-đun trong số chúng cung cấp các chức năng như tập tin I/O (vào/ra), các lệnh gọi hàm hệ thống, các socket, và thậm chí các giao tiếp với các công cụ giao diện đồ họa như Tk.

- Python là một ngôn ngữ thông dịch, điều đó giúp tiết kiệm thời gian trong quá trình phát triển chương trình vì việc biên dịch hay liên kết là không cần thiết. Bộ thông dịch có thể được dùng một cách tương tác, làm cho việc thử nghiệm các tính năng của ngôn ngữ trở nên dễ dàng. Trình thông dịch thực thi code theo từng dòng, điều này giúp cho quá trình debug trở nên dễ dàng hơn và đây cũng là yếu tố khá quan trọng giúp Python thu hút được nhiều người học và trở nên khá phổ biến.
- Python cũng là một ngôn ngữ lập trình hướng đối tượng. Ngoài ra, Python còn hỗ trợ các phương thức lập trình theo hàm và theo cấu trúc.
- Python cho phép viết các chương trình nhỏ gọn và dễ hiểu. Các chương trình viết bằng Python thường ngắn hơn so với các chương trình viết bằng C, C++ hoặc Java.
- Ngoài các đặc điểm trên, Python còn khá nhiều đặc điểm khác như hỗ trợ lập trình GUI, mã nguồn mở, có thể tích hợp với các ngôn ngữ lập trình khác, …

1.2.3. Một số thư viện liên quan.

Python có hỗ trợ rất nhiều thư viện, sau đây là một số thư viện phổ biến và được sử dụng trong chương demo thuật toán, các định nghĩa được tham khảo tại tài liệu số [2]:

- Numpy là một thư viện không thể thiếu khi chúng ta xây dựng các ứng dụng Máy học trên Python. NumPy là một cấu trúc dữ liệu tổng quát, đại số tuyến tính và thư viện thao tác ma trận cho Python. Trong numpy, chiều của mảng gọi là axes, trong khi số chiều gọi là rank.
- Pandas là một trong những thư viện được dùng rộng rãi nhất trong Python cùng với Numpy. Pandas cung cấp nhiều đối tượng và phương thức cho các cấu trúc dữ liệu. Pandas là thư viện không thể thiếu cho chúng ta trong suốt quá trình xử lý dữ liệu, từ chuyển đổi hay ánh xạ dữ liệu thô sang dạng dữ liệu mà chúng ta mong muốn, nhằm có thể phân tích dễ dàng hơn. Pandas

là một thư viện phần mềm để phân tích dữ liệu dạng bảng và dữ liệu chuỗi thời gian. Theo nhiều cách, nó tái tạo chức năng của đối tượng R's DataFrame. Ngoài ra, nhiều tính năng phổ biến của Microsoft Excel có thể được thực hiện bằng cách sử dụng Pandas, chẳng hạn như "nhóm theo", trục bảng, xóa và chèn cột dễ dàng.

Các đối tượng DataFrame của Pandas dựa trên nhãn (trái ngược với dựa trên chỉ mục như trường hợp của NumPy), do đó mỗi cột thường được đặt một tên có thể được gọi để thực hiện các hoạt động. Các đối tượng DataFrame giống với bảng tính hơn và mỗi cột của DataFrame có thể có một kiểu khác nhau, chẳng hạn như boolean, số hoặc văn bản. Thông thường, nó sẽ được nhấn mạnh, bạn sẽ sử dụng NumPy và Pandas kết hợp. Hai thư viện bổ sung cho nhau và không phải là các khuôn khổ cạnh tranh, mặc dù có sự chồng chéo về chức năng giữa hai thư viện. Đầu tiên, chúng ta phải lấy một số dữ liệu. Gói Python SciKit-Learn cung cấp một số dữ liệu mẫu mà chúng ta có thể sử dụng.

- Scikit-learn là một thư viện mã nguồn mở dành cho học máy - một ngành trong trí tuệ nhân tạo, rất mạnh mẽ và thông dụng với cộng đồng Python, được thiết kế trên nền NumPy và SciPy. Scikit-learn cung cấp giao diện chuẩn hóa cho nhiều thuật toán máy học được sử dụng phổ biến nhất và là thư viện phổ biến nhất và thường xuyên được sử dụng để học máy cho Python. Ngoài việc cung cấp nhiều thuật toán học tập, SciKit-Learn có một số lượng lớn các chức năng tiện lợi cho các tác vụ tiền xử lý thông thường (ví dụ: chuẩn hóa hoặc xác nhận chéo k-lần). Scikit-learn là một thư viện phần mềm rất lớn, đi kèm với documentations, luôn được cập nhật.

1.3. HTML.

1.3.1. Giới thiệu về HTML.

HTML là chữ viết tắt của cụm từ HyperText Markup Language (dịch là Ngôn ngữ đánh dấu siêu văn bản) được sử dụng để tạo một trang web, trên một website có thể sẽ chứa nhiều trang và mỗi trang được quy ra là một tài liệu HTML (thi thoảng mình sẽ ghi là một tập tin HTML). Cha đẻ của HTML là Tim Berners-Lee, cũng là người khai sinh ra World Wide Web và chủ tịch của World Wide Web Consortium (W3C – tổ chức thiết lập ra các chuẩn trên môi trường Internet).

Một tài liệu HTML được hình thành bởi các phần tử HTML (HTML Elements) được quy định bằng các cặp thẻ (tag), các cặp thẻ này được bao bọc bởi một dấu ngoặc nhọn (ví dụ <html>) và thường là sẽ được khai báo thành một cặp, bao gồm thẻ mở và thẻ đóng (ví dụ và). Các văn bản muốn được đánh dấu bằng HTML sẽ được khai báo bên trong cặp thẻ (ví dụ Đây là chữ in đậm). Nhưng một số thẻ đặc biệt lại không có thẻ đóng và dữ liệu được khai báo sẽ nằm trong các thuộc tính (ví dụ như thẻ <img>).

Một tập tin HTML sẽ bao gồm các phần tử HTML và được lưu lại dưới đuôi mở rộng là .html hoặc .htm.

1.3.2. Vai trò của HTML.

HTML là một ngôn ngữ đánh dấu siêu văn bản nên nó sẽ có vai trò xây dựng cấu trúc siêu văn bản trên một website, hoặc khai báo các tập tin kỹ thuật số (media) như hình ảnh, video, nhạc.

Điều đó không có nghĩa là chỉ sử dụng HTML để tạo ra một website mà HTML chỉ đóng một vai trò hình thành trên website.

HTML – Xây dựng cấu trúc và định dạng các siêu văn bản.

CSS – Định dạng các siêu văn bản dạng thô tạo ra từ HTML thành một bố cục website, có màu sắc, ảnh nền…

Javascript – Tạo ra các sự kiện tương tác với hành vi của người dùng (ví dụ nhấp vào ảnh trên nó sẽ có hiệu ứng phóng to).

Dễ hiểu hơn, bạn hãy nghĩ rằng nếu website là một cơ thể hoàn chỉnh thì HTML chính là bộ xương của cơ thể đó, nó như là một cái khung sườn vậy.

Như vậy, dù website thuộc thể loại nào, giao tiếp với ngôn ngữ lập trình nào để xử lý dữ liệu thì vẫn phải cần HTML để hiển thị nội dung ra cho người truy cập xem.

1.4. CSS.

1.4.1. Giới thiệu về CSS.

CSS là chữ viết tắt của Cascading Style Sheets, nó chỉ đơn thuần là một dạng file text với phần tên mở rộng là .css. Trong Style Sheet này chứa những câu lệnh CSS. Mỗi một lệnh của CSS sẽ định dạng một phần nhất định của HTML ví dụ như: font của chữ, đường viền, màu nền, căn chỉnh hình ảnh…

Trước đây khi chưa có CSS, những người thiết kế web phải trộn lẫn giữa các thành phần trình bày và nội dung với nhau. Nhưng với sự xuất hiện của CSS, người ta

có thể tách rời hoàn toàn phần trình bày và nội dung. Giúp cho phần code của trang web cũng gọn hơn và quan trọng hơn cả là dễ chỉnh sửa hơn.

CSS được phát triển bởi W3C (World Wide Web Consortium) vào năm 1996, vì một lý do đơn giản. HTML không được thiết kế để gắn tag để giúp định dạng trang web. Bạn chỉ có thể dùng nó để "đánh dấu" lên site.

Những tag như <font> được ra mắt trong HTML phiên bản 3.2, nó gây rất nhiều rắc rối cho lập trình viên. Vì website có nhiều font khác nhau, màu nền và phong cách khác nhau. Để viết lại code cho trang web là cả một quá trình dài, cực nhọc. Vì vậy, CSS được tạo bởi W3C là để giải quyết vấn đề này.

Mối tương quan giữa HTML và CSS rất mật thiết. HTML là ngôn ngữ markup (nền tảng của site) và CSS định hình phong cách (tất cả những gì tạo nên giao diện website), chúng là không thể tách rời.

CSS về lý thuyết không có cũng được, nhưng khi đó website sẽ chỉ là một trang chứa văn bản mà không có gì khác.

CSS cho phép tạo các quy tắc chỉ định cách nội dung của một phần tử sẽ xuất hiện. Ví dụ: có thể chỉ định rằng nền của trang là màu kem, tất cả các đoạn văn sẽ xuất hiện bằng màu xám bằng kiểu chữ Arial hoặc tất cả các tiêu đề cấp một phải có màu xanh lam, in nghiêng, kiểu chữ Times (có thể tham khảo Chương 10 trong cuốn "Html&Css" [6] của John Ducket).

1.4.2. Ưu điểm của CSS.

Sự khác biệt giữa site có CSS và không có CSS rất dễ nhận biết.

Trước khi sử dụng CSS, tất cả những phong cách của CSS cần được đính kèm vào trong HTML markup. Có nghĩa là bạn cần tách ra để xác định các thành phần như background, font colors, canh hàng…

CSS giúp định kiểu mọi thứ trên một file khác, bạn có thể tạo phong cách trước rồi sau đó tích hợp file CSS lên trên cùng của file HTML. Việc này giúp HTML markup rõ ràng và dễ quản lý hơn nhiều.

Tóm lại, với CSS bạn không cần lặp lại các mô tả cho từng thành phần. Nó tiết kiệm thời gian, làm code ngắn lại để bạn có thể kiểm soát lỗi dễ dàng hơn.

1.5. JavaScript.

1.5.1. Giới thiệu về JavaScript.

JavaScript là ngôn ngữ lập trình phổ biến nhất trên thế giới trong suốt 20 năm qua. Nó cũng là một trong số 3 ngôn ngữ chính của lập trình web:

- HTML: Giúp bạn thêm nội dung cho trang web.
- CSS: Định dạng thiết kế, bố cục, phong cách, canh lề của trang web.
- JavaScript: Cải thiện cách hoạt động của trang web.

Chi tiết cụ thể hơn có thể tham khảo trong tài liệu [7].

1.5.2. Ưu điểm của JavaScript.

JavaScript có rất nhiều ưu điểm khiến nó vượt trội hơn so với các đối thủ, đặc biệt trong các trường hợp thực tế. Sau đây chỉ là một số lợi ích của JavaScript:

Bạn không cần một compiler vì web browser có thể biên dịch nó bằng HTML.

- Nó dễ học hơn các ngôn ngữ lập trình khác.
- Lỗi dễ phát hiện hơn và vì vậy dễ sửa hơn.
- Nó có thể được gắn trên một số element của trang web hoặc event của trang web như là thông qua click chuột hoặc di chuột tới.
- JS hoạt động trên nhiều trình duyệt, nền tảng…
- Bạn có thể sử dụng JavaScript để kiểm tra input và giảm thiểu việc kiểm tra thủ công khi truy xuất qua cơ sở dữ liệu.
- Nó giúp website tương tác tốt hơn với khách truy cập.
- Nó nhanh hơn và nhẹ hơn các ngôn ngữ lập trình khác.

1.5.3. Nhược điểm của JavaScript.

Mọi ngôn ngữ lập trình đều có các khuyết điểm. Một phần là vì ngôn ngữ đó khi phát triển đến một mức độ như JavaScript, nó cũng sẽ thu hút lượng lớn hacker, scammer, và những người có ác tâm luôn tìm kiếm những lỗ hổng và các lỗi bảo mật để lợi dụng nó. Một số khuyết điểm có thể kể đến là:

- Dễ bị khai thác.
- Có thể được dùng để thực thi mã độc trên máy tính của người dùng.
- Nhiều khi không được hỗ trợ trên mọi trình duyệt.
- Có thể bị triển khai khác nhau tùy từng thiết bị dẫn đến việc không đồng nhất.

1.6. Bootstrap.

1.6.1. Giới thiệu về Bootstrap.

Bootstrap là một framework bao gồm các HTML, CSS và JavaScript template dùng để phát triển website chuẩn responsive. Bootstrap cho phép quá trình thiết kế website diễn ra nhanh chóng và dễ dàng hơn dựa trên những thành tố cơ bản sẵn có như typography, forms, buttons, tables, grids, navigation, image carousels…

Bootstrap là một bộ sưu tập miễn phí của các mã nguồn mở và công cụ dùng để tạo ra một mẫu webiste hoàn chỉnh. Với các thuộc tính về giao diện được quy định sẵn như kích thước, màu sắc, độ cao, độ rộng…, các designer có thể sáng tạo nhiều sản phẩm mới mẻ nhưng vẫn tiết kiệm thời gian khi làm việc với framework này trong quá trình thiết kế giao diện website.

1.6.2. Lý do chọn Bootstrap.

Bootstrap rất phổ biến và là một lựa chọn tối ưu trong thiết kế web.

Giữa muôn vàn ứng dụng thiết kế website hiện nay, Bootstrap vẫn có khả năng cạnh tranh cao là nhờ những đặc điểm nổi bật sau:

- Dễ dàng thao tác.

  Cơ chế hoạt động của Bootstrap là dựa trên xu hướng mã nguồn mở HTML, CSS và Javascript. Người dùng cần trang bị kiến thức cơ bản 3 mã này mới có thể sử dụng Bootstrap hiệu quả. Bên cạnh đó, các mã nguồn này cũng có thể dễ dàng thay đổi và chỉnh sửa tùy ý.

- Tùy chỉnh dễ dàng.

  Bootstrap được tạo ra từ các mã nguồn mở cho phép designer linh hoạt hơn. Giờ đây có thể lựa chọn những thuộc tính, phần tử phù hợp với dự án họ đang theo đuổi. CDN Boostrap còn giúp bạn tiết kiệm dung lượng vì không cần tải mã nguồn về máy.

- Chất lượng sản phẩm đầu ra hoàn hảo.

  Bootstrap là sáng tạo của các lập trình viên giỏi trên khắp thế giới. Bootstrap đã được nghiên cứu và thử nghiệm trên các thiết bị. Được kiểm tra nhiều lần trước khi đưa vào sử dụng. Do đó, khi chọn Bootstrap, bạn có thể tin rằng mình sẽ tạo nên những sản phẩm với chất lượng tốt nhất.

- Độ tương thích cao.

  Điểm cộng lớn nhất của Bootstrap là khả năng tương thích với mọi trình duyệt và nền tảng. Đây là một điều cực kì quan trọng và cần thiết trong trải nghiệm người dùng. Sử dụng Grid System cùng với hai bộ tiền xử lý Less và Sass, Bootstrap mặc định hỗ trợ Responsive và ưu tiên cho các giao diện trên thiết bị di động hơn. Bootstrap có khả năng tự động điều chỉnh kích thước trang website theo khung browser. Mục đích để phù hợp với màn hình của máy tính để bàn, tablet hay laptop.

1.6.3. *Cấu trúc và tính năng của Bootstrap.*

Cấu trúc gọn nhẹ khiến chức năng của Bootstrap trở nên linh hoạt.

Bootstrap chứa các tập tin JavaScript, CSS và fonts đã được biên dịch và nén lại. Ngoài ra, Bootstrap được thiết kế dưới dạng các mô-đun. Do đó, dễ dàng tích hợp với hầu hết các mã nguồn mở như WordPress, Joomla, Magento, …Trong đó, Bootstrap mang đến nhiều chức năng nổi bật.

- Bootstrap cho phép người dùng truy cập vào thư viện "khổng lồ" các thành tố dùng để tạo nên giao diện của một website hoàn chỉnh như font, typography, form, table, grid…
- Bootstrap cho phép bạn tùy chỉnh framework của website trước khi tải xuống và sử dụng nó tại trang web của khung.
- Tái sử dụng các thành phần lặp đi lặp lại trên trang web.
- Bootstrap được tích hợp jQuery. Bạn chỉ cần khai báo chính xác các tính năng trong quá trình lập trình web của bạn.
- Định nghĩa glyphicons nhằm giảm thiểu việc sử dụng hình ảnh làm biểu tượng và tăng tốc độ tải trang.

1.7. Kết luận chương 1.

Trong chương này, tôi đã giới thiệu về tổng quan Machine Learning, phân nhóm các loại thuật toán, cũng như trình bày các bước khi thực hiện một dự án với ML. Ngoài ra, tôi còn giới thiệu về các ngôn ngữ Python, HTML, CSS, JavaScript, Bootstrap và lý do tại sao lại lựa chọn chúng.

Hiểu HTML và CSS có thể giúp ích cho bất kỳ ai làm việc với web; các nhà thiết kế có thể tạo ra các trang web hấp dẫn hơn và có thể sử dụng được, người chỉnh sửa trang web có thể tạo nội dung tốt hơn, các nhà tiếp thị có thể giao tiếp với khán giả của

họ hiệu quả hơn và người quản lý có thể giao các trang web tốt hơn và tận dụng tối đa nhóm của họ.

Tôi sẽ bắt đầu đi vào phân tích và đánh giá thuật toán KNN ở chương 2 tiếp theo.

CHƯƠNG 2. PHÂN TÍCH – ĐÁNH GIÁ

2.1. Sơ lược về K-Nearest Neighbors.

Nếu như con người có kiểu học "nước đến chân mới nhảy", thì trong Machine Learning cũng có một thuật toán như vậy.

2.1.1. Giới thiệu về K-Nearest Neighbors.

K-nearest neighbor là một trong những thuật toán supervised-learning đơn giản nhất (mà hiệu quả trong một vài trường hợp) trong Machine Learning. Khi training, thuật toán này không học một điều gì từ dữ liệu training (đây cũng là lý do thuật toán này được xếp vào loại lazy learning), mọi tính toán được thực hiện khi nó cần dự đoán kết quả của dữ liệu mới. K-nearest neighbor có thể áp dụng được vào cả hai loại của bài toán Supervised learning là Classification và Regression. KNN còn được gọi là một thuật toán Instance-based hay Memory-based learning.

Để dễ hiểu hơn, hãy thử xem một câu chuyện vui sau:

Có một anh bạn chuẩn bị đến ngày thi cuối kỳ. Vì môn này được mở tài liệu khi thi nên anh ta không chịu ôn tập để hiểu ý nghĩa của từng bài học và mối liên hệ giữa các bài. Thay vào đó, anh thu thập tất cả các tài liệu trên lớp, bao gồm ghi chép bài giảng (lecture notes), các slides và bài tập về nhà + lời giải. Để cho chắc, anh ta ra thư viện và các quán Photocopy quanh trường mua hết tất cả các loại tài liệu liên quan (khá khen cho cậu này chịu khó tìm kiếm tài liệu). Cuối cùng, anh bạn của chúng ta thu thập được một chồng cao tài liệu để mang vào phòng thi.

Vào ngày thi, anh tự tin mang chồng tài liệu vào phòng thi. Aha, đề này ít nhất mình phải được 8 điểm. Câu 1 giống hệt bài giảng trên lớp. Câu 2 giống hệt đề thi năm ngoái mà lời giải có trong tập tài liệu mua ở quán Photocopy. Câu 3 gần giống với bài tập về nhà. Câu 4 trắc nghiệm thậm chí cậu nhớ chính xác ba tài liệu có ghi đáp án. Câu cuối cùng, 1 câu khó nhưng anh đã từng nhìn thấy, chỉ là không nhớ ở đâu thôi.[6]

Kết quả cuối cùng, cậu ta được 4 điểm, vừa đủ điểm qua môn. Cậu làm chính xác câu 1 vì tìm được ngay trong tập ghi chú bài giảng. Câu 2 cũng tìm được đáp án nhưng lời giải của quán Photocopy sai! Câu ba thấy gần giống bài về nhà, chỉ khác mỗi một số thôi, cậu cho kết quả giống như thế luôn, vậy mà không được điểm nào. Câu 4 thì tìm được cả 3 tài liệu nhưng có hai trong đó cho đáp án A, cái còn lại cho B. Cậu chọn A và được điểm. Câu 5 thì không làm được dù còn tới 20 phút, vì tìm mãi chẳng thấy đáp án đâu - nhiều tài liệu quá cũng mệt!!

Thuật toán KNN khá giống với cách học/thi của anh bạn kém may mắn kia.

Có một vài khái niệm tương ứng người-máy như sau:

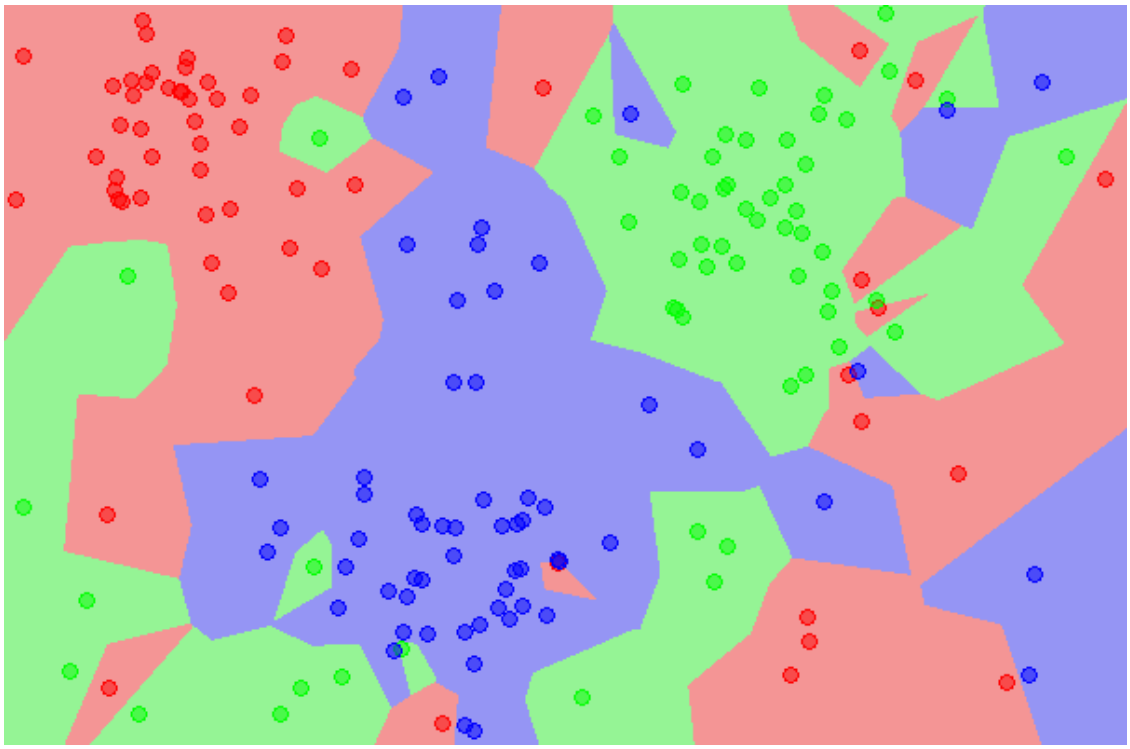
Bảng 2. 1 Khái niệm tương ứng người-máy

Ngôn ngữ người	**Ngôn ngữ Máy Học**	**Trong Machine Learning**
Câu hỏi	Điểm dữ liệu	Data point
Đáp án	Đầu ra, nhãn	Output, Label
Ôn thi	Huấn luyện	Training
Tập tài liệu mang vào phòng thi	Tập dữ liệu tập huấn	Training set
Đề thi	Tập dữ liểu kiểm thử	Test set
Câu hỏi trong đề thi	Dữ liệu kiểm thử	Test data point
Câu hỏi có đáp án sai	Nhiễu	Noise, Outlier
Câu hỏi gần giống	Điểm dữ liệu gần nhất	Nearest Neighbor

Với KNN, trong bài toán Classification, label của một điểm dữ liệu mới (hay kết quả của câu hỏi trong bài thi) được suy ra trực tiếp từ K điểm dữ liệu gần nhất trong training set. Label của một test data có thể được quyết định bằng major voting (bầu chọn theo số phiếu) giữa các điểm gần nhất, hoặc nó có thể được suy ra bằng cách đánh trọng số khác nhau cho mỗi trong các điểm gần nhất đó rồi suy ra label.

Trong bài toán Regresssion, đầu ra của một điểm dữ liệu sẽ bằng chính đầu ra của điểm dữ liệu đã biết gần nhất (trong trường hợp K=1), hoặc là trung bình có trọng số của đầu ra của những điểm gần nhất, hoặc bằng một mối quan hệ dựa trên khoảng cách tới các điểm gần nhất đó.

Một cách ngắn gọn, KNN là thuật toán đi tìm đầu ra của một điểm dữ liệu mới bằng cách chỉ dựa trên thông tin của K điểm dữ liệu trong training set gần nó nhất (K-lân cận), không quan tâm đến việc có một vài điểm dữ liệu trong những điểm gần nhất này là nhiễu. Hình dưới đây là một ví dụ về KNN trong classification với K = 1.



Hình 2. 1 Bản đồ 1NN

Ví dụ trên đây là bài toán Classification với 3 classes: Đỏ, Lam, Lục. Mỗi điểm dữ liệu mới (test data point) sẽ được gán label theo màu của điểm mà nó thuộc về. Trong hình này, có một vài vùng nhỏ xen lẫn vào các vùng lớn hơn khác màu. Ví dụ có một điểm màu Lục ở gần góc 11 giờ nằm giữa hai vùng lớn với nhiều dữ liệu màu Đỏ và Lam. Điểm này rất có thể là nhiễu. Dẫn đến nếu dữ liệu test rơi vào vùng này sẽ có nhiều khả năng cho kết quả không chính xác.

2.1.2. Khoảng cách trong không gian vector.

Trong không gian một chiều, khoảng cách giữa hai điểm là trị tuyệt đối giữa hiệu giá trị của hai điểm đó. Trong không gian nhiều chiều, khoảng cách giữa hai điểm có thể được định nghĩa bằng nhiều hàm số khác nhau, trong đó độ dài đường thẳng nối hai điểm chỉ là một trường hợp đặc biệt trong đó.

2.2. Phân tích ý tưởng.

Ý tưởng:

Hãy cho tôi biết bạn của bạn là ai, tôi sẽ cho bạn biết bạn là người như thế nào.

Đó là một câu danh ngôn mà có lẽ ai cũng biết. Và có lẽ, phần lớn đều đồng ý. Đây là ý tưởng đằng sau thuật toán KNN. Ý tưởng là ghi nhớ tập huấn luyện và sau đó dự đoán nhãn của bất kỳ điểm dữ liệu mới nào trên cơ sở nhãn của những điểm hàng xóm gần nhất của nó trong tập huấn luyện. Cơ sở lý luận của phương pháp như vậy dựa

trên giả định rằng các đối tượng địa lý được sử dụng để mô tả các điểm miền có liên quan đến nhãn của chúng theo cách làm cho các điểm gần nhau có thể có cùng nhãn. Hơn nữa, trong một số tình huống, ngay cả khi tập hợp đào tạo là rất lớn, việc tìm một hàng xóm gần nhất có thể được thực hiện cực kỳ nhanh chóng.

Ví dụ, để dự đoán mẫu dữ liệu mới xnew thuộc về lớp nào, ta dựa vào số k dữ liệu gần nó nhất. Ví dụ k=3, nghĩa là gần xnew có 3 điểm dữ liệu. Giả sử trong đó có 2 điểm dữ liệu thuộc về lớp B và 1 điểm dữ liệu thuộc về lớp A. Như vậy, ta sẽ gán xnew thuộc về lớp B do xnew có 2 điểm dữ liệu thuộc lớp B gần nó nhất.

Thuật toán:

- Thuật toán xác định lớp cho mẫu mới xnew.
- Tính khoảng cách giữa xnew và tất cả các mẫu trong tập huấn luyện.
- Chọn k mẫu gần nhất với xnew trong tập huấn luyện.
- Gán xnew vào lớp có nhiều mẫu nhất trong số k mẫu láng giềng đó (hoặc xnew nhận giá trị trung bình của k mẫu).

Lưu ý rằng, phương pháp Nearest Neighbor tìm ra nhãn trên bất kỳ điểm kiểm tra nào mà không cần tìm kiếm công cụ dự đoán trong một số lớp hàm được xác định trước.

2.3. Quy trình thiết kế thuật toán.

Các bước thực hiện thuật toán có thể đơn giản như sau:

- Chuẩn bị dữ liệu (dữ liệu đã được làm sạch, chuyển đổi, sẵn sàng đưa vào phân tích), chia tập dữ liệu ra làm 2: training data set (để train model) và test data set (để kiểm chứng model).
- Chọn một số K bất kỳ, K là một số nguyên, tức là số điểm dữ liệu đã phân loại có khoảng cách gần nhất (láng giềng gần nhất) với điểm dữ liệu chưa phân loại.
- Tính toán khoảng cách giữa điểm dữ liệu chưa phân loại với các điểm dữ liệu đã được phân loại.
- Với kết quả có được sắp xếp theo thứ tự với giá trị khoảng cách từ bé nhất đến lớn nhất.
- Chọn ra các điểm dữ liệu có giá trị khoảng cách bé nhất với điểm dữ liệu cần phân loại dựa trên K cho trước, ví dụ nếu K = 2 tức chọn ra 2 điểm dữ liệu gần nhất, K = 3 là 3 điểm dữ liệu gần nhất.

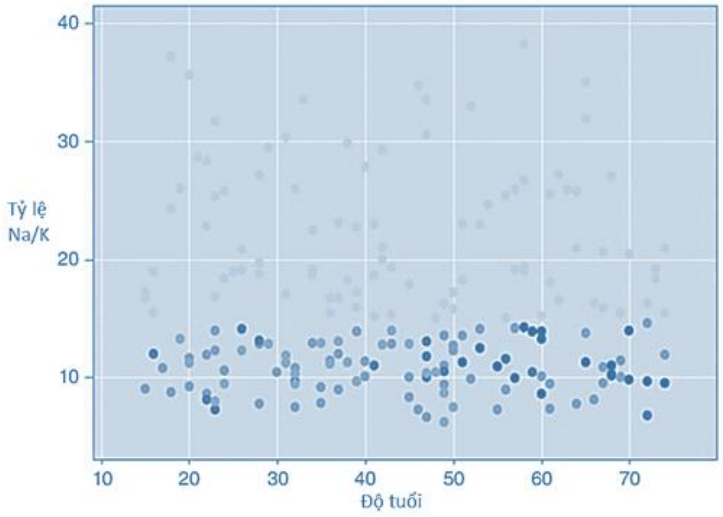
- Tiếp theo xem xét giá trị của biến mục tiêu (biến phân loại) của các điểm dữ liệu gần nhất, chọn ra giá trị xuất hiện nhiều nhất và gán cho điểm dữ liệu chưa phân loại, ví dụ K = 3, trong đó có 2 điểm dữ liệu được phân loại là A, điểm còn lại là B thì điểm dữ liệu chưa phân loại lúc này sẽ được phân loại là A.
- Kiểm chứng lại độ hiệu quả của model trên test data set, và sử dụng các phương pháp đánh giá khác nhau.
- Thay đổi giá trị K khác nhau và thực hiện lại quy trình để tìm được K tối ưu nhất cho tập dữ liệu.

Bước khó khăn nhất của thuật toán KNN, và cũng là bước đau đầu nhất, cần sự kinh nghiệm của nhà phân tích, đó chính là chọn K là bao nhiêu. Nhưng trước hết chúng ta cùng đi qua ví dụ cụ thể sau đây để biết được cách tính khoảng cách giữa các điểm dữ liệu.

2.4. Ví dụ minh hoạ.

Ví dụ sau được tham khảo từ giáo trình Data mining nổi tiếng: "*Discovering Knowledge in Data: An Introduction to Data Mining*" của Daniel T.Larose (có thể tham khảo tài liệu tham khảo [4]):

Giả sử một bệnh viện tiến hành phân loại thuốc chỉ định cho những bệnh nhân mới dựa trên độ tuổi (Age) và tỷ lệ Na/K trong máu.



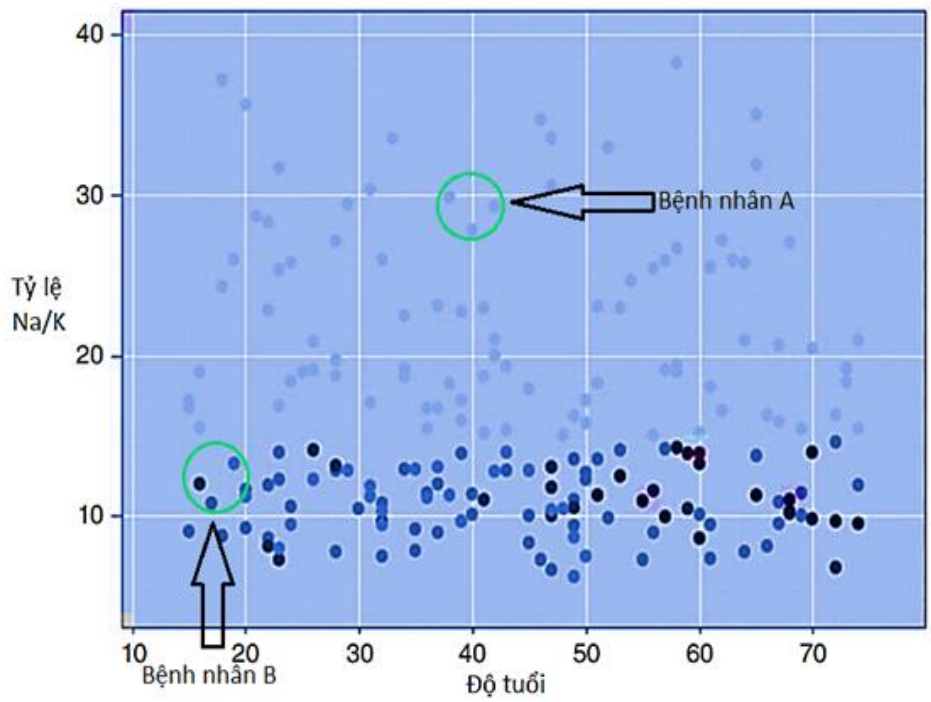
Hình 2. 2 Đồ thị Scatter Plot

Trục hoành là độ tuổi, trục tung là tỷ lệ Na/K, mỗi điểm trên đồ thị là một bệnh nhân tương ứng với tỷ lệ Na/K, và độ tuổi cho trước. Màu sắc khác nhau thể hiện cho loại thuốc chỉ định.

Màu xanh nhạt là loại thuốc M, màu xanh trung bình là loại thuốc N, màu xanh đậm là loại thuốc P.

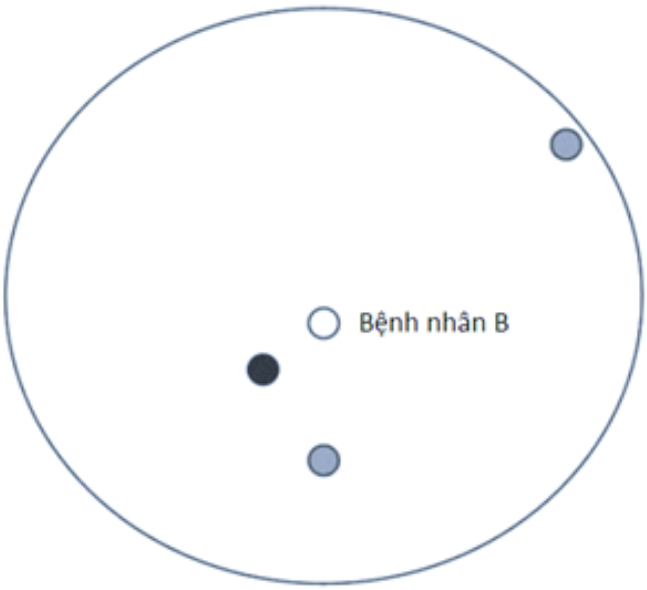
Giả sử bệnh viện tiếp nhận các bệnh nhân mới (ví dụ bệnh nhân A và B) và cần tiến hành phân loại thuốc cho họ. Đồ thị tiếp theo dưới đây chứa các bệnh nhân mới chưa được phân loại, dựa vào độ tuổi, và tỷ lệ Na/K chúng ta xác định được những vùng trên đồ thị sẽ là nơi chứa các data point của các bệnh nhân mới này. Nhiệm vụ là xác định loại thuốc thích hợp cho bệnh nhân A, B trên cơ sở là xác định khoảng cách giữa điểm dữ liệu của A, và B (chưa được phân loại thuốc) và điểm dữ liệu đã phân loại (bệnh nhân cũ trước đây đã được phân loại thuốc), khoảng cách gần nhất thì khả năng loại thuốc được phân loại sẽ tương đương nhau giữa 2 bệnh nhân.

Trước tiên xét bệnh nhân A giả sử có độ tuổi là 40 và tỷ lệ Na/K gần 29, thì thấy rằng điểm dữ liệu của bệnh nhân này nằm trong vùng chứa có các điểm dữ liệu màu xanh nhạt tức nằm chung vùng với các bệnh nhân trước đây được phân loại thuốc là M. Do đó bệnh nhân mới số 1 sẽ được phân loại thuốc là M. Ở đây không cần đặt giá trị K để tìm ra các điểm gần nhất do xung quanh của điểm dữ liệu bệnh nhân A toàn là các điểm màu xanh nhạt.



Hình 2. 3 Đồ thị được điều chỉnh màu sắc để nhìn rõ các điểm

Xét tiếp bệnh nhân B, lưu ý hình đã được cân chỉnh lại màu sắc để hiển thị rõ màu sắc khác nhau giữa các điểm giúp các bạn dễ phân biệt.



Hình 2. 4 Hình phóng lại gần vùng chứa điểm dữ liệu của bệnh nhân B

Nếu lấy K = 1 tức chỉ xét 1 điểm gần nhất so với điểm dữ liệu của bệnh nhân B, thì điểm dữ liệu bệnh nhân B gần nhất với điểm màu xanh đậm nhất, ứng với bệnh nhân B sẽ được phân loại thuốc là P. Nếu chúng ta lấy K = 2 tức xét 2 điểm gần nhất, thì điểm dữ liệu B sẽ gần với 1 điểm xanh đậm và 1 điểm màu xanh trung bình, tức là bệnh nhân B có thể được phân loại thuốc là P hoặc là N. Do đó chưa tìm ra đâu là loại thuốc thích hợp nhất cho B, vậy K = 2 không phải là giá trị K cần xét. Tiếp đến lấy K = 3, thì trên đồ thị chúng ta thấy có 2 điểm màu xanh trung bình nhiều hơn so với 1 điểm màu xanh đậm là gần nhất với điểm dữ liệu bệnh nhân B. Vậy với K = 3, bệnh nhân B sẽ được phân loại thuốc là N khi điểm dữ liệu bệnh nhân B gần với nhiều điểm dữ liệu màu xanh trung bình hơn.

Phương pháp trên gọi là Voting, tức tìm ra những điểm dữ liệu phổ biến xuất hiện gần nhất với điểm dữ liệu cần phân loại (trong thống kê cũng có thể gọi là tính Mode). Trở lại ví dụ trên thì với K = 3, số vote cho điểm dữ liệu màu xanh trung bình là 2, còn điểm dữ liệu màu xanh đậm là 1, vậy 2>1, nên bệnh nhân B sẽ được phân loại thuốc là P, độ tin cậy Confidence = 2/3 = 66.7%.

Lưu ý quan trọng, thường tìm K trước rồi mới khoanh vùng cho điểm dữ liệu chưa phân loại dựa trên việc tính toán các khoảng cách giữa nó so với các điểm dữ liệu đã phân loại. Khoanh vùng trước để dễ dàng trình bày ví dụ mà thôi.

Vậy thì việc tính khoảng cách giữa các điểm dữ liệu dựa trên 2 phương pháp chính:

- Euclidean:

$$d_{\text{Euclidean}}(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_i (x_i - y_i)^2}$$

- Manhattan:

$$d_{Manhattan}(\mathbf{x}, \mathbf{y}) = \sum_i \left| x_i - y_i \right|$$

Trong bài báo cáo này tôi sẽ chỉ dùng công thức Euclidean phổ biến nhất để tính toán các khoảng cách giữa các điểm dữ liệu.

Ví dụ bệnh nhân mới D có tuổi là 20, và tỷ lệ Na/K là 12, bệnh nhân cũ E có tuổi là 30, tỷ lệ Na/K là 8. Vậy khoảng cách $d_{euclidean} = \sqrt{(28-35)^2 + (11-9)^2} = 7.28$.

Trong quá trình tính toán khoảng cách sẽ có những biến chứa các giá trị lớn ví dụ như tiền và những biến chứa giá trị nhỏ ví dụ như độ tuổi, giá trị các khoảng cách được tính sẽ không còn phù hợp và hiệu quả. Lúc này các chuyên gia phân tích thường phải chuẩn hóa dữ liệu.

Chuẩn hóa dữ liệu Z-score:

$$z = \frac{x - \bar{x}}{s}$$

Chuẩn hóa dữ liệu Min – max:

$$X^* = \frac{X - \min(X)}{\text{range}(X)} = \frac{X - \min(X)}{\max(X) - \min(X)}$$

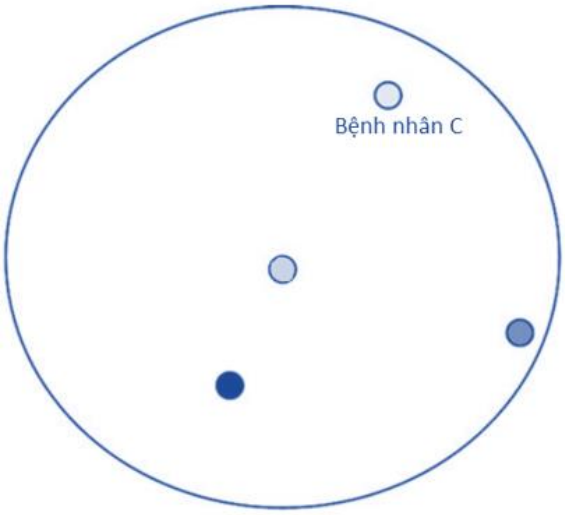
Giá trị chuẩn hóa Z nằm trong khoảng từ -3 đến 3 (theo quy tắc thực nghiệm Empirical Rule trong thống kê) còn giá trị chuẩn hóa Min – Max từ 0 đến 1 do trong độ trải giữa (range) chứa giá trị X. Chuẩn hóa theo Min – Max thường được ưa chuộng hơn khi các biến đầu vào có thêm biến định tính.

Trong trường hợp biến đầu vào phục vụ cho việc phân loại không phải là biến định lượng như độ tuổi hay tỷ lệ Na/K mà là biến định tính ví dụ như tình trạng hôn nhân (độc thân, đã kết hôn), hay giới tính (nam, nữ), chúng ta sẽ dùng quy tắc sau để tính khoảng cách:

$X_i – Y_i = 0$ nếu có cùng giá trị của biến định tính và bằng 1 nếu ngược lại.

Quay trở lại với cách thức chọn K vừa đề cập ở trên là cách thức chọn K đơn giản, hay còn gọi là Simple Unweighted Voting tức mỗi một điểm dữ liệu ở gần điểm

dữ liệu cần phân loại sẽ được 1 vote. Ví dụ ở trên khi K = 3 xét cho bệnh nhân B, thì số điểm dữ liệu có màu xanh trung bình là 2, điểm dữ liệu màu xanh đậm là 1 vậy bệnh nhân B được phân loại thuốc theo các điểm màu xanh trung bình, là thuốc P. Tuy nhiên nếu giả sử bệnh nhân C là bệnh nhân mới, cũng với K = 3, nhưng khi tính toán lại tìm ra 3 điểm gần với C nhất nhưng mỗi một điểm tương ứng mỗi màu khác nhau, tức màu xanh đậm, xanh trung bình, và xanh nhạt.



Hình 2. 5 Có lượng vote ngang nhau

Dưới góc nhìn của các chuyên gia thì những điểm dữ liệu hay data point gần với data point của dữ liệu cần phân loại thì cần có giá trị vote cao hơn, do đó cần sử dụng phương pháp Weighted voting với giá trị vote được tính bằng cách nghịch đảo giá trị bình phương của khoảng cách (trên cơ sở giá trị của dữ liệu đã được chuẩn hóa).

Ngoài ra, trong quá trình xem xét các thuộc tính, hay biến đầu vào, các chuyên gia cho rằng những biến nào có tác động nhiều hơn, có ảnh hưởng nhiều hơn đến biến mục tiêu (biến phân loại) cần được gán trọng số. Dựa trên kinh nghiệm phân tích, cùng với sử dụng các phương pháp trong Data mining, và dưới góc nhìn của mình thì các chuyên gia sẽ có cách thức xác định trọng số khác nhau.

Giả sử ví dụ về phân loại thuốc, các bác sĩ thấy rằng tỷ lệ Na/K quan trọng hơn độ tuổi trong việc phân loại, do đó gán trọng số là 3 và độ tuổi là 1. Tính lại khoảng cách giữa bệnh nhân mới D và bệnh nhân cũ E đã được phân loại thuốc trước đó, chúng ta có:

$$D_{euclidean} = \sqrt{(28-35)^2 + 3*(11-9)^2} = 7.8$$

Như vậy, tôi đã trình bày tổng quan về những kiến thức cơ bản trong việc tính toán khoảng cách giữa 2 điểm dữ liệu, và cách thức tính vote để tìm ra giá trị phân loại chính xác cho đối tượng dữ liệu chưa được phân loại.

Tiếp theo trở lại với vấn đề ban đầu là nên xác định giá trị K sao cho phù hợp, tức là cần bao nhiêu điểm ở gần nhất hay số "láng giềng" gần nhất với điểm dữ liệu cần phân loại?

Đầu tiên, theo lời khuyên của các chuyên gia thì nên có phép thử, nghĩa là chạy nhiều mô hình KNN với các giá trị K khác nhau và bắt đầu thử nghiệm từ K = 1, sau đó kiểm tra độ hiệu quả của từng mô hình (dựa trên các phương pháp đánh giá mô hình phân loại mà chúng tôi sẽ giới thiệu ở các bài viết sau), mô hình nào hiệu quả nhất thì giá trị K sẽ tối ưu. Nguyên nhân là do mỗi tập dữ liệu khác nhau sẽ có các tính chất, đặc điểm khác nhau, nên khó có thể xác định được giá trị K nào là phù hợp nhất.

Để hình dung rõ ràng hơn về cách nó tính, tôi sẽ giới thiệu thêm 1 ví dụ có số lượng thuộc tính phức tạp hơn.

Giả sử một ngân hàng có một tập dữ liệu gồm 5000 khách hàng đã mở các khoản vay tiêu dùng khác nhau, sau một khoản thời gian cụ thể, ngân hàng đã xác định được có 3895 khách hàng thanh toán khoản vay đúng hạn và có 1105 khách hàng không thanh toán khoản vay đúng hạn, nếu xét nợ quá hạn là sau 3 tháng kể từ thời điểm thanh toán toàn bộ, thì các khách hàng này đang trong tình trạng nợ quá hạn. Ngân hàng sẽ tổng hợp toàn bộ dữ liệu và phân loại các khách hàng này theo có khả năng nợ xấu và không có khả năng nợ xấu.

Đặt Y là khả năng nợ xấu, với Y = 0 là không có khả năng nợ xấu, ngược lại là Y = 1.

Với các biến đầu vào X_i bao gồm ví dụ các biến sau:

- Giới tính (nam, nữ).
- Độ tuổi.
- Thu nhập hàng tháng (trong ví dụ này sẽ không xét đến nghề nghiệp do quá đa dạng).
- Trình độ học vấn.
- Tình trạng hôn nhân.
- Tình trạng sở hữu bất động sản.
- Khoản vay.

- Thời hạn vay (tháng).

Giả định là lãi suất là như nhau đối với các khoản vay tiêu dùng (vay thế chấp).

Nhân viên ngân hàng lấy mẫu 15 khách hàng đã được phân loại khả năng nợ xấu để phân tích cho một khách hàng mới mở khoản vay.

ID	Độ tuổi	Giới tính	Học vấn	Hôn nhân	Sở hữu BĐS	Thu nhập	Khoản vay	Thời hạn vay	Nợ xấu
1	25	Nam	ĐH	Độc thân	Ở cùng bố mẹ	7000000	20000000	6	0
2	40	Nam	THPT	Đã kết hôn	Nhà sở hữu	20000000	100000000	9	0
3	35	Nữ	ĐH	Từng ly hôn	Nhà thuê	12000000	30000000	4	1
4	27	Nam	THPT	Đã kết hôn	Ở cùng bố mẹ	9000000	15000000	6	0
5	31	Nữ	THCS	Độc thân	Nhà thuê	6000000	23000000	6	1
6	36	Nữ	ĐH	Đã kết hôn	Nhà sở hữu	8000000	20000000	6	1
7	48	Nam	ĐH	Độc thân	Nhà thuê	7000000	10000000	3	0
8	26	Nữ	ĐH	Đã kết hôn	Nhà thuê	8000000	10000000	3	1
9	33	Nam	CĐ	Từng ly hôn	Ở cùng bố mẹ	5000000	12000000	4	1
10	29	Nam	THPT	Độc thân	Nhà thuê	10000000	25000000	9	0
11	38	Nam	THPT	Đã kết hôn	Nhà sở hữu	15000000	60000000	9	0
12	44	Nữ	ĐH	Độc thân	Nhà sở hữu	14000000	20000000	4	1
13	42	Nam	ĐH	Đã kết hôn	Nhà sở hữu	10000000	30000000	4	0
14	28	Nữ	ĐH	Độc thân	Nhà thuê	7000000	20000000	6	1
15	30	Nữ	ĐH	Đã kết hôn	Ở cùng bố mẹ	6000000	16000000	6	1
16	32	Nam	ĐH	Đã kết hôn	Nhà sở hữu	15000000	40000000	6	????

Hình 2. 6 Mẫu 15 khách hàng và 1 khách hàng mới

Sau đây, tôi sẽ chuẩn hóa dữ liệu theo Min-max:

Bảng 2. 2 Chuẩn hóa dữ liệu Min-max Thu nhập

ID	Thu nhập	Chuẩn hóa dữ liệu	Đã chuẩn hóa
1	7000000	$\frac{7000000-5000000}{20000000-5000000}$	0.13
2	20000000	$\frac{20000000-5000000}{20000000-5000000}$	1
3	12000000	$\frac{12000000-5000000}{20000000-5000000}$	0.47
4	9000000	$\frac{9000000-5000000}{20000000-5000000}$	0.27
5	6000000	$\frac{6000000-5000000}{20000000-5000000}$	0.07
6	8000000	$\frac{8000000-5000000}{20000000-5000000}$	0.2
7	7000000	$\frac{7000000-5000000}{20000000-5000000}$	0.13
8	8000000	$\frac{8000000-5000000}{20000000-5000000}$	0.2
9	5000000	$\frac{5000000-5000000}{20000000-5000000}$	0

ID	Thu nhập	Chuẩn hóa dữ liệu	Đã chuẩn hóa
10	10000000	$\frac{10000000-5000000}{20000000-5000000}$	0.33
11	15000000	$\frac{15000000-5000000}{20000000-5000000}$	0.67
12	14000000	$\frac{14000000-5000000}{20000000-5000000}$	0.6
13	10000000	$\frac{10000000-5000000}{20000000-5000000}$	0.33
14	7000000	$\frac{7000000-5000000}{20000000-5000000}$	0.13
15	6000000	$\frac{6000000-5000000}{20000000-5000000}$	0.07
16	15000000	$\frac{15000000-5000000}{20000000-5000000}$	0.67

Bảng 2. 3 Chuẩn hóa dữ liệu Min-max Khoản vay

ID	Khoản vay	Chuẩn hóa dữ liệu	Đã chuẩn hóa
1	20000000	$\frac{20000000-10000000}{100000000-10000000}$	0.11
2	100000000	$\frac{100000000-10000000}{100000000-10000000}$	1
3	30000000	$\frac{30000000-10000000}{100000000-10000000}$	0.22
4	15000000	$\frac{15000000-10000000}{100000000-10000000}$	0.06
5	23000000	$\frac{23000000-10000000}{100000000-10000000}$	0.14
6	20000000	$\frac{20000000-10000000}{100000000-10000000}$	0.11
7	10000000	$\frac{10000000-10000000}{100000000-10000000}$	0
8	10000000	$\frac{10000000-10000000}{100000000-10000000}$	0
9	12000000	$\frac{12000000-10000000}{100000000-10000000}$	0.02
10	250000000	$\frac{25000000-10000000}{100000000-10000000}$	0.17

ID	Khoản vay	Chuẩn hóa dữ liệu	Đã chuẩn hóa
11	60000000	$\frac{60000000-10000000}{100000000-10000000}$	0.56
12	20000000	$\frac{20000000-10000000}{100000000-10000000}$	0.11
13	30000000	$\frac{30000000-10000000}{100000000-10000000}$	0.22
14	20000000	$\frac{20000000-10000000}{100000000-10000000}$	0.11
15	16000000	$\frac{16000000-10000000}{100000000-10000000}$	0.07
16	40000000	$\frac{40000000-10000000}{100000000-10000000}$	0.33

Bảng 2. 4 Chuẩn hóa dữ liệu Min-max Thời hạn vay

ID	Thời hạn vay	Chuẩn hóa dữ liệu	Đã chuẩn hóa
1	6	$\frac{6-3}{9-3}$	0.5
2	9	$\frac{9-3}{9-3}$	1
3	4	$\frac{4-3}{9-3}$	0.17
4	6	$\frac{6-3}{9-3}$	0.5
5	6	$\frac{6-3}{9-3}$	0.5
6	6	$\frac{6-3}{9-3}$	0.5
7	3	$\frac{3-3}{9-3}$	0
8	3	$\frac{3-3}{9-3}$	0
9	4	$\frac{4-3}{9-3}$	0.17
10	9	$\frac{9-3}{9-3}$	1
11	9	$\frac{9-3}{9-3}$	1

ID	Thời hạn vay	Chuẩn hóa dữ liệu	Đã chuẩn hóa
12	4	$\frac{4-3}{9-3}$	0.17
13	4	$\frac{4-3}{9-3}$	0.17
14	6	$\frac{6-3}{9-3}$	0.5
15	6	$\frac{6-3}{9-3}$	0.5
16	6	$\frac{6-3}{9-3}$	0.5

- Chuẩn hóa dữ liệu Min-max Độ tuổi tính tương tự.

Chúng ta chuẩn hóa dữ liệu định lượng theo phương pháp Min – max để có được bảng dữ liệu như sau:

ID	Độ tuổi	Giới tính	Học vấn	Hôn nhân	Sở hữu BĐS	Thu nhập	Khoản vay	Thời hạn vay	Nợ xấu
1	0.00	Nam	ĐH	Độc thân	Ở cùng bố mẹ	0.13	0.11	0.50	0
2	0.65	Nam	THPT	Đã kết hôn	Nhà sở hữu	1.00	1.00	1.00	0
3	0.43	Nữ	ĐH	Từng ly hôn	Nhà thuê	0.47	0.22	0.17	1
4	0.09	Nam	THPT	Đã kết hôn	Ở cùng bố mẹ	0.27	0.06	0.50	0
5	0.26	Nữ	THCS	Độc thân	Nhà thuê	0.07	0.14	0.50	1
6	0.48	Nữ	ĐH	Đã kết hôn	Nhà sở hữu	0.20	0.11	0.50	1
7	1.00	Nam	ĐH	Độc thân	Nhà thuê	0.13	0.00	0.00	0
8	0.04	Nữ	ĐH	Đã kết hôn	Nhà thuê	0.20	0.00	0.00	1
9	0.35	Nam	CĐ	Từng ly hôn	Ở cùng bố mẹ	0.00	0.02	0.17	1
10	0.17	Nam	THPT	Độc thân	Nhà thuê	0.33	0.17	1.00	0
11	0.57	Nam	THPT	Đã kết hôn	Nhà sở hữu	0.67	0.56	1.00	0
12	0.83	Nữ	ĐH	Độc thân	Nhà sở hữu	0.60	0.11	0.17	1
13	0.74	Nam	ĐH	Đã kết hôn	Nhà sở hữu	0.33	0.22	0.17	0
14	0.13	Nữ	ĐH	Độc thân	Nhà thuê	0.13	0.11	0.50	1
15	0.22	Nữ	ĐH	Đã kết hôn	Ở cùng bố mẹ	0.07	0.07	0.50	1
16	0.30	Nam	ĐH	Đã kết hôn	Nhà sở hữu	0.67	0.33	0.50	????

Hình 2. 7 Dữ liệu đã được chuẩn hóa

Tiếp theo tôi sẽ tính khoảng cách giữa các điểm dữ liệu, lưu ý ở các biến định tính, nếu giá trị bằng nhau thì sẽ là 0, ngược lại là 1:

Vì công thức quá dài, khi viết công thức căn bậc 2 trong báo cáo không thể xuống dòng, nên tôi chỉ nêu ví dụ khoảng cách tính từ ID 16 (điểm đang xét) đến ID 1 (điểm dữ liệu đầu tiên):

$$d\ (ID\ 16, ID\ 1) = \sqrt{(0.3-0)^2+0^2+0^2+1^2+1^2+(0.67-0.13)^2+(0.33-0.11)^2+(0.5-0.5)^2}$$

$$= 1.557.$$

ID	d (ID 16, ID …)	Xếp hạng	Nợ xấu
1	1.557	6	0
2	1.389	4	0
3	1.784	11	1
4	1.511	5	0
5	2.097	15	1
6	1.140	1	1
7	1.771	10	0
8	1.626	9	1
9	1.912	14	1
10	1.845	13	0
11	1.170	2	0
12	1.563	8	1
13	1.195	3	0
14	1.834	12	1
15	1.561	7	1

Hình 2. 8 Các giá trị khoảng cách từ ID 16 (điểm xét) đến các điểm còn lại

Bên trên là các giá trị khoảng cách tính được qua công thức Euclidean, và xếp hạng với khoảng cách bé nhất tức là gần nhất được xếp hạng là 1.

Nếu chọn K = 1, thì khách hàng mới có ID 16 sẽ được phân loại khả năng nợ xấu theo khách hàng ID 6 là 1, nếu K = 2, tức chọn ra 2 điểm gần nhất tức chọn ra được ID 6 và ID 11 nhưng mỗi điểm lại có 2 giá trị khả năng nợ xấu khác nhau nên không phân loại được cho ID 16, nếu K = 3 thì chúng ta chọn được 3 điểm gần nhất là ID 13, ID 11, ID 6, trong đó 2 điểm có khả năng nợ xấu là 0, vậy ID 16 được phân loại khả năng nợ xấu là 0.

	Khách hàng	Nợ xấu = 0	Nợ xấu = 1
K = 1	ID 6	0	1
K = 2	ID 6, ID 11	1	1
K = 3	ID 6, ID 11, ID 13	2	1
K = 4	ID 6, ID 11, ID 13, ID 2	3	1
K = 5	ID 6, ID 11, ID 13, ID 2, ID 4	4	1
K = 6	ID 6, ID 11, ID 13, ID 2, ID 4, ID 1	5	1
K = 7	ID 6, ID 11, ID 13, ID 2, ID 4, ID 1, ID 15	5	2

Hình 2. 9 Chọn K và giá trị vote

Bên trên là cách thức chọn K và giá trị vote có được cho Nợ xấu = 0, Nợ xấu = 1, ví dụ K = 7, có 7 điểm gần nhất so với ID 16 là ID 6, ID 11, ID 13, ID 2, ID 4, ID 1, ID 15, trong đó có 5 khách hàng được phân loại nợ xấu bằng 0, có 2 khách hàng được phân loại nợ xấu bằng 1. Vậy ID 16 sẽ được phân loại nợ xấu là 0.

Trong thực tế, như đã nói, việc chọn K phù hợp phải dựa trên việc chạy thử mô hình KNN với từng giá trị K khác nhau và đánh giá độ hiệu quả của từng mô hình.

Nếu chọn K = 2, thì phải sử dụng phương pháp Weighted Voting tức tính nghịch đảo bình phương của d(ID 16, ID 6) và nghịch đảo bình phương của d(ID 16, ID 11) rồi so sánh 2 giá trị với nhau, giá trị nào lớn hơn thì ID 16 sẽ được phân loại nợ xấu theo khách hàng ấy.

Như vậy tôi đã minh họa cách thức thực hiện một thuật toán KNN đơn giản nhất ứng dụng trong ngân hàng để dự báo khả năng khách hàng mới có thể không trả kịp khoản vay đúng hạn. Tuy trong thực tế, phải xét rất nhiều biến khác nhau tác động lên khả năng nợ xấu, và phân tích một khối lượng lớn dữ liệu, sử dụng nhiều phương pháp khác nhau để có được kết quả chính xác nhưng KNN vẫn được coi là một trong những cách tiếp cận hiệu quả.

2.5. Demo thuật toán.

Ta sẽ sử dụng thuật toán KNN của scikit-learn để phân lớp dữ liệu với tập huấn luyện là Iris. Tập dữ liệu hoa Iris được Ronald Fisher giới thiệu vào năm 1936, một tập dữ liệu cổ điển trong máy học và thống kê. Tập dữ liệu này được dùng cho các bài toán phân lớp. Dữ liệu gồm 50 mẫu thu thập từ ba loại hoa Iris (Iris setosa, Iris virginica và Iris versicolor). Bốn thuộc tính cho mỗi mẫu gồm chiều dài và chiều rộng của đài hoa và cánh hoa được tính theo đơn vị centimet.

In [1]:

```
# k-Nearest Neighbor
from sklearn import datasets
from sklearn import metrics
from sklearn.neighbors import KneighborsClassifier
# tải bộ dữ liệu iris
dataset = datasets.load_iris()
# 6 dòng dữ liệu đầu tiên của tập dữ liệu Iris
dataset.data[0:6]
```

Out[1]:

```
array([[5.1, 3.5, 1.4, 0.2],
       [4.9, 3. , 1.4, 0.2],
```

```
[4.7, 3.2, 1.3, 0.2],
[4.6, 3.1, 1.5, 0.2],
[5. , 3.6, 1.4, 0.2],
[5.4, 3.9, 1.7, 0.4]])
```

Tiếp theo, ta sẽ cài đặt chương trình minh hoạ cho thuật toán KNN.

In [2]:

```
# làm cho mô hình k-nearest neighbor phù hợp dữ liệu
model = KNeighborsClassifier()
model.fit(dataset.data, dataset.target)
print(model)
KNeighborsClassifier(algorithm='auto', leaf_size=30,
metric='minkowski',
        metric_params=None, n_jobs=1, n_neighbors=5, p=2,
        weights='uniform')
```

In [3]:

```
# dự đoán
expected = dataset.target
predicted = model.predict(dataset.data)
# sự phù hợp của mô hình
print(metrics.classification_report(expected, predicted))
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	50
1	0.96	0.94	0.95	50
2	0.94	0.96	0.95	50
avg / total	0.97	0.97	0.97	150

In [4]:

```
print(metrics.confusion_matrix(expected, predicted))
```

```
[[50  0  0]
 [ 0 47  3]
 [ 0  2 48]]
```

Qua ví dụ trên, ta thấy thuật toán KNN ứng dụng khá tốt trên tập dữ liệu Iris. KNN cho độ chính xác (precision) trung bình là 97%, (recall) trung bình là 97%, và (f1-score) là 97%. Để cài đặt thuật toán, ta chỉ cần khai báo lớp KNeighborsClassifier và gọi hàm fit() ứng với mô hình phân lớp này.

2.6. Ưu và nhược điểm của K-Nearest Neighbors.

Ưu điểm của KNN:

- Độ phức tạp tính toán của quá trình training là bằng 0.
- Việc dự đoán kết quả của dữ liệu mới rất đơn giản.
- Không cần giả sử gì về phân phối của các class.

Nhược điểm của KNN:

- KNN rất nhạy cảm với nhiễu khi K nhỏ.
- Như đã nói, KNN là một thuật toán mà mọi tính toán đều nằm ở khâu test. Trong đó việc tính khoảng cách tới từng điểm dữ liệu trong training set sẽ tốn rất nhiều thời gian, đặc biệt là với các cơ sở dữ liệu có số chiều lớn và có nhiều điểm dữ liệu. Với K càng lớn thì độ phức tạp cũng sẽ tăng lên. Ngoài ra, việc lưu toàn bộ dữ liệu trong bộ nhớ cũng ảnh hưởng tới hiệu năng của KNN.

2.7. Kết luận chương 2.

Trong chương này, tôi đã giới thiệu tổng quan về thuật toán K-Nearest Neighbors, cách thuật toán hoạt động, cũng như cách xây dựng thuật toán và nêu ra các hạn chế của thuật toán. Do đó, KNN thường được thực hiện trong thực tế sau bước tiền xử lý giảm kích thước.

Chương tiếp theo, tôi sẽ ứng dụng thuật toán vào xây dựng website dự đoán tình trạng hôn nhân.

CHƯƠNG 3. ỨNG DỤNG THUẬT TOÁN K-NEAREST NEIGHBORS VÀO BÀI TOÁN DỰ ĐOÁN TÌNH TRẠNG HÔN NHÂN

3.1. Phát biểu bài toán.

Những năm gần đây, tình trạng ly hôn, li dị ngày một gia tăng ở các gia đình, năm sau cao hơn năm trước, trong đó phần lớn là những người còn trẻ. Theo một nghiên cứu tỷ lệ ly hôn ở Việt Nam đang tăng nhanh và chiếm 31%-40%, hàng năm có hàng chục ngàn trẻ em phải chịu cảnh thiếu cha mẹ do ly hôn. Ly hôn gia tăng đã và đang để lại hệ lụy cho người trong cuộc, cho thế hệ trẻ và cho xã hội.

Có thể nói, ly hôn là sự lựa chọn của đàn ông đàn bà phụ nữ với nhau, nhưng hệ quả của nó có tác động đến tâm sinh lý của những đứa trẻ; để lại gánh nặng cho xã hội nếu như con cái của họ bị bỏ rơi, không được chăm sóc, nuôi dưỡng, giáo dục chu đáo; chúng sẽ thiếu đi sự chăm sóc, tình cảm của người cha hoặc người mẹ, thậm chí cả hai. Từ đó, sẽ ảnh hưởng tới quá trình phát triển nhân cách, dễ sa ngã vào những tệ nạn xã hội... Đây cũng là một trong những lý do mà trong những năm gần đây tình trạng tội phạm tuổi vị thành niên có xu hướng gia tăng.

Kết hôn rồi ly hôn, dễ đến rồi lại dễ đi. Lý do để ra đi thì nhiều lắm thế mà không thể tìm lấy một lý do để ở lại. Khoảnh khắc hai người cùng kí vào tờ đơn ly hôn cũng chính là lúc họ đã tước đi mái ấm, phá bỏ đi hạnh phúc gia đình của chính những đứa con của họ.

Do đó, dự doán là để có giải pháp tránh cho tình trạng ly hôn xảy ra hay còn để cải thiện mối quan hệ.

3.2. Tập dữ liệu sử dụng.

Trước khi xây dựng một mô hình học máy, thường nên kiểm tra dữ liệu, để xem liệu nhiệm vụ có thể giải quyết dễ dàng mà không cần máy học hay không hoặc liệu thông tin mong muốn có thể không được chứa trong dữ liệu hay không.

Ngoài ra, kiểm tra dữ liệu của bạn là một cách tốt để tìm ra những điểm bất thường và đặc biệt. Trong thế giới thực, sự không nhất quán trong dữ liệu và các phép đo không mong muốn là rất phổ biến.

Tập dữ liệu lấy từ UCIMachinelearning tại tài liệu tham khảo [8] và nó cung cấp tất cả thông tin liên quan cần thiết cho việc dự đoán Ly hôn. Nó chứa 54 đặc điểm và trên cơ sở những đặc điểm này, chúng ta phải dự đoán rằng cặp đôi đã ly hôn hay chưa.

Giá trị 1 đại diện cho Đã ly hôn và giá trị 0 đại diện cho không ly hôn. Các đặc điểm gồm:

1. Nếu một trong số chúng tôi xin lỗi khi cuộc thảo luận của chúng tôi xấu đi, cuộc thảo luận kết thúc.
2. Tôi biết chúng ta có thể bỏ qua sự khác biệt của mình, ngay cả khi đôi khi mọi thứ trở nên khó khăn.
3. Khi chúng tôi cần, chúng tôi có thể trao đổi với vợ / chồng của tôi ngay từ đầu và sửa nó.
4. Khi tôi tranh cãi với vợ/chồng tôi, cuối cùng tôi sẽ liên lạc với cô/anh ấy.
5. Thời gian tôi dành cho vợ/chồng thật đặc biệt đối với chúng tôi.
6. Chúng tôi gần như không bao giờ có thời gian ở nhà như là cặp đôi.
7. Chúng tôi giống như hai người xa lạ có chung môi trường ở nhà hơn là gia đình.
8. Tôi tận hưởng kỳ nghỉ của chúng tôi với vợ/chồng tôi.
9. Tôi thích đi du lịch cùng vợ/chồng.
10. Hầu hết các mục tiêu của chúng tôi là chung cho vợ / chồng tôi.
11. Tôi nghĩ rằng một ngày nào đó trong tương lai, khi nhìn lại, tôi thấy rằng vợ / chồng tôi và tôi đã hòa hợp với nhau.
12. Vợ/chồng tôi và tôi có những giá trị tương đồng nhau về quyền tự do cá nhân.
13. Vợ chồng tôi có những trò giải trí giống nhau.
14. Hầu hết các mục tiêu của chúng ta đối với mọi người (trẻ em, bạn bè, v.v.) đều giống nhau.
15. Ước mơ của chúng tôi được sống với vợ/chồng tôi rất giống nhau và hòa hợp.
16. Chúng tôi tương thích với vợ/chồng của tôi về tình yêu nên là gì.
17. Chúng tôi có cùng quan điểm về việc hạnh phúc trong cuộc sống của chúng tôi với vợ/chồng của tôi.
18. Vợ/chồng tôi và tôi có những suy nghĩ giống nhau về cuộc hôn nhân nên như thế nào.
19. Vợ/chồng tôi và tôi có những ý kiến tương tự nhau về vai trò nên có trong hôn nhân.

20. Vợ tôi và tôi có những giá trị tương tự về sự tin tưởng.
21. Tôi biết chính xác những gì vợ/chồng tôi thích.
22. Tôi biết vợ/chồng của tôi muốn được chăm sóc như thế nào khi cô ấy / anh ấy bị ốm.
23. Tôi biết món ăn yêu thích của vợ / chồng tôi.
24. Tôi có thể nói cho bạn biết loại căng thẳng mà vợ / chồng tôi đang phải đối mặt trong cuộc sống của cô ấy / anh ấy.
25. Tôi có kiến thức về thế giới nội tâm của vợ / chồng tôi.
26. Tôi biết những lo lắng cơ bản của vợ / chồng tôi.
27. Tôi biết nguồn căng thẳng hiện tại của vợ / chồng tôi là gì.
28. Tôi biết hy vọng và mong muốn của vợ / chồng tôi.
29. Tôi biết vợ/chồng của mình rất tốt.
30. Tôi biết bạn bè của vợ / chồng tôi và các mối quan hệ xã hội của họ.
31. Tôi cảm thấy hung hăng khi cãi nhau với vợ / chồng.
32. Khi thảo luận với vợ/chồng của tôi, tôi thường sử dụng các thành ngữ như 'Bạn luôn luôn' hoặc 'Bạn không bao giờ'.
33. Tôi có thể sử dụng những tuyên bố tiêu cực về tính cách của vợ / chồng tôi trong các cuộc thảo luận của chúng tôi.
34. Tôi có thể sử dụng các biểu hiện xúc phạm trong các cuộc thảo luận của chúng tôi.
35. Tôi có thể xúc phạm vợ/chồng của mình trong các cuộc thảo luận của chúng tôi.
36. Tôi có thể bị sỉ nhục khi chúng tôi thảo luận.
37. Cuộc tranh cãi của tôi với vợ/chồng của tôi không được bình tĩnh.
38. Tôi ghét cách vợ / chồng của tôi mở một chủ đề.
39. Những trận đánh nhau thường xảy ra đột ngột.
40. Chúng tôi chỉ bắt đầu một cuộc chiến trước khi tôi biết chuyện gì đang xảy ra.
41. Khi tôi nói chuyện với vợ/chồng của mình về điều gì đó, sự bình tĩnh của tôi đột nhiên bị phá vỡ.
42. Khi tôi tranh cãi với vợ, nó chỉ nổ ra và tôi không nói một lời.
43. Tôi chủ yếu muốn làm dịu môi trường một chút.

44. Đôi khi tôi nghĩ rằng thật tốt cho tôi khi tôi rời khỏi nhà một thời gian.
45. Tôi thà im lặng hơn là cãi vã với vợ / chồng tôi.
46. Ngay cả khi tôi đúng trong cuộc tranh luận, tôi khát khao không làm mất lòng bên kia.
47. Khi tôi thảo luận với vợ/chồng của mình, tôi im lặng vì tôi sợ không thể kiềm chế cơn giận của mình.
48. Tôi cảm thấy đúng trong các cuộc thảo luận của chúng tôi.
49. Tôi không liên quan gì đến những gì tôi đã bị buộc tội.
50. Tôi thực sự không phải là người có tội về những gì tôi bị buộc tội.
51. Tôi không phải là người sai về các vấn đề ở nhà.
52. Tôi sẽ không ngần ngại nói với vợ / chồng của tôi về sự bất cập của cô ấy/anh ấy.
53. Khi trao đổi, tôi nhắc nhở cô/anh ấy về những vấn đề chưa thỏa đáng của vợ/chồng tôi.
54. Tôi không ngại nói với vợ / chồng mình về sự kém cỏi của anh ấy / cô ấy.

Bản thân tập dữ liệu chứa 170 hàng và 55 cột. 54 cột đầu đại diện cho 54 đặc điểm có giá trị từ 0-4, cột thứ 55 (cột cuối cùng) đại diện cho nhãn có giá trị 0 và 1.

3.3. Phương pháp tiếp cận.

Ta có một bộ dữ liệu về 54 thuộc tính và nhãn của từng đối tượng là tập dữ liệu mẫu. Với 170 điểm dữ liệu (đối tượng), ta tách ra làm 2 tập, một tập huấn luyện để chạy thuật toán và một tập kiểm thử để đánh giá độ chính xác của thuật toán. Khi có 1 điểm dữ liệu mới, ta tính khoảng cách từ điểm mới đó đến các điểm trong bộ dữ liệu mẫu, từ đó lấy ra k điểm gần nhất. Trong k điểm đó, ta tiến hành bỏ phiếu để xác định nhãn nào có số lượng nhiều hơn (cụ thể trong đề tài này sẽ có 2 lớp nhãn là 0: Chưa ly hôn và 1: Đã ly hôn). Kết quả dự đoán chính là kết quả bỏ phiếu.

3.4. Xây dựng thuật toán.

Sau đây, tôi sẽ xây dựng mô hình học máy KNN.

3.4.1. Thiết kế.

Để xây dựng thuật toán, ở đây ta sử dụng một số thư viện của Python như: numpy, pandas, sklearn,... đã được trình bày ở phần trên.

Đầu tiên, ta import các thư viện cần dùng:

```
import numpy as np
from sklearn import neighbors
import pandas as pd
```

Hình 3. 1 Câu lệnh import các thư viện cần dùng

Tiếp theo, chúng ta nhập dữ liệu vào:

```
xl = pd.ExcelFile('data\divorce.xlsx')
# get the first sheet as an object
duLieu = pd.read_excel(xl, 0, header=None)
divorce = duLieu.values[1:]
```

Hình 3. 2 Nhập dữ liệu từ file xlsx

Và rồi, ta tách thuộc tính nhãn và 54 đặc điểm ra riêng:

```
divorce_y = divorce[:,-1]          # target value is the last column
divorce_y = divorce_y.astype('int')
divorce_X = divorce[:,0:-1]        # features are the other columns
```

Hình 3. 3 Lấy các nhãn và 54 đặc điểm

Sau khi hoàn tất phần chuẩn bị dữ liệu, thì tiến hành phân tách dữ liệu.

Scikit-learning chứa một hàm xáo trộn tập dữ liệu và phân tách nó: hàm train_test_split. Hàm này trích xuất 75% số hàng trong dữ liệu dưới dạng tập huấn luyện, cùng với các nhãn tương ứng cho dữ liệu này. 25% dữ liệu còn lại, cùng với các nhãn còn lại, được khai báo là tập kiểm tra. Quyết định bao nhiêu dữ liệu muốn đưa vào đào tạo và tập kiểm tra tương ứng là một số tùy ý, nhưng sử dụng tập kiểm tra chứa 25% dữ liệu là một nguyên tắc chung.

Trong scikit-learning, dữ liệu thường được ký hiệu bằng chữ X viết hoa, trong khi các nhãn được ký hiệu bằng chữ y viết thường. Điều này được lấy cảm hứng từ công thức tiêu chuẩn f (x) = y trong toán học, trong đó x là đầu vào của một hàm và y là đầu ra. Theo các quy ước khác từ toán học, chúng tôi sử dụng chữ X viết hoa vì dữ liệu là mảng hai chiều (ma trận) và chữ y viết thường vì mục tiêu là mảng một chiều (vectơ).

Trước khi thực hiện tách, hàm train_test_split xáo trộn tập dữ liệu bằng cách sử dụng trình tạo số ngẫu nhiên. Nếu chỉ lấy 25% dữ liệu cuối cùng làm tập kiểm tra, tất cả các điểm dữ liệu sẽ có nhãn 1, vì các điểm dữ liệu được sắp xếp theo nhãn. Việc sử dụng bộ thử nghiệm chỉ chứa một trong hai lớp sẽ không cho biết nhiều về mô hình tổng quát

tốt như thế nào, vì vậy xáo trộn dữ liệu để đảm bảo dữ liệu thử nghiệm chứa dữ liệu từ tất cả các lớp.

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    divorce_X, divorce_y, test_size=56)
```

Hình 3. 4 Phân tách dữ liệu test và train

Mã trên chạy mô hình kNN trên tập dữ liệu huấn luyện gồm **X_train** đại diện cho tập 54 thuộc tính của tình trạng hôn nhân và **Y_train** đại diện cho kết quả ly hôn hay không ly hôn. Code trả về một giá trị đánh giá điểm của mô hình khi chạy trên tập kiểm tra.

Đoạn code bên trên phân chia tập dữ liệu thành 2 phần tương ứng **67%** cho training và **33%** cho testing.

Bây giờ có thể bắt đầu xây dựng mô hình học máy thực tế.

Tất cả các mô hình học máy trong scikit-learning đều được triển khai trong các lớp riêng của chúng, được gọi là các lớp Estimator. Thuật toán phân lớp kNN được thực hiện trong lớp KNeighborsClassifier trong mô-đun neighbors. Trước khi có thể sử dụng mô hình, chúng ta cần khởi tạo lớp thành một đối tượng. Đây là lúc chúng ta sẽ thiết lập bất kỳ thông số nào của mô hình. Tham số quan trọng nhất của KNeighborsClassifier là số lượng hàng xóm, chúng tôi sẽ đặt thành 10:

```
clf = neighbors.KNeighborsClassifier(n_neighbors = 10, p = 2)
```

Hình 3. 5 Khởi tạo lớp thành một đối tượng

Đối tượng knn đóng gói thuật toán sẽ được sử dụng để xây dựng mô hình từ dữ liệu huấn luyện, cũng như thuật toán để đưa ra dự đoán về các điểm dữ liệu mới. Nó cũng sẽ giữ thông tin mà thuật toán đã trích xuất từ dữ liệu huấn luyện. Trong trường hợp của KNeighborsClassifier, nó sẽ chỉ lưu trữ tập huấn luyện.

Để xây dựng mô hình trên tập huấn luyện, chúng ta gọi phương thức fit của đối tượng knn, phương thức này nhận các đối số là mảng NumPy X_train chứa dữ liệu huấn luyện và mảng NumPy y_train của các nhãn huấn luyện tương ứng:

```
clf.fit(X_train, y_train)
```

Hình 3. 6 Xây dựng mô hình trên tập huấn luyện

Phương thức fit trả về bản thân đối tượng knn (và sửa đổi nó tại chỗ), vì vậy kết quả nhận được một biểu diễn chuỗi của bộ phân lớp.

```
>>> clf.fit(X_train, y_train)
KNeighborsClassifier(n_neighbors=10)
```

Hình 3. 7 Kết quả hiển thị của phương thức fit

Bây giờ có thể đưa ra dự đoán bằng cách sử dụng mô hình này trên dữ liệu mới mà có thể không biết các nhãn chính xác.

Để đưa ra dự đoán, gọi phương thức predict của đối tượng knn:

```
y_pred = clf.predict(X_test)
```

Hình 3. 8 Dự đoán tình trạng hôn nhân của tất cả các điểm kiểm thử

Thử xuất ra kết quả dự đoán 20 điểm bất kì trong bộ kiểm thử, so với kết quả gốc của nó để có cái nhìn trực quan về độ chính xác của thuật toán:

```
print("\n\nPrint results for 20 test data points:")
print("Predicted labels: ", y_pred[20:40])
print("Ground truth    : ", y_test[20:40])
```

Hình 3. 9 Lấy ra kết quả dự đoán 20 điểm bất kì trong bộ dữ liệu kiểm thử

```
Print results for 20 test data points:
Predicted labels:  [0 1 0 0 0 0 0 1 0 1 1 1 1 1 0 0 0 0 1 1]
Ground truth    :  [0 1 0 0 0 0 0 1 0 1 1 1 1 1 1 0 0 0 1 1]
```

Hình 3. 10 Nhãn dự đoán và nhãn gốc của 20 điểm bất kì trong bộ dữ liệu kiểm thử

Cuối cùng, ta đánh giá độ chính xác của thuật toán:

```
from sklearn.metrics import accuracy_score
print("\n\nAccuracy of 10NN: %.2f %%" %(100*accuracy_score(y_test, y_pred)))
```

Hình 3. 11 Đánh giá độ chính xác thuật toán

Phương thức accuracy_score của đối tượng knn, phương pháp này sẽ tính toán độ chính xác của bộ kiểm tra.

Source code đầy đủ:

```
import numpy as np
from sklearn import neighbors
import pandas as pd

xl = pd.ExcelFile('data\divorce.xlsx')
# get the first sheet as an object
duLieu = pd.read_excel(xl, 0, header=None)
divorce = duLieu.values[1:]

divorce_y = divorce[:,-1]         # target value is the last column
divorce_y = divorce_y.astype('int')
divorce_X = divorce[:,0:-1]       # features are the other columns

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    divorce_X, divorce_y, test_size=56)
clf = neighbors.KNeighborsClassifier(n_neighbors = 10, p = 2)
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
print("\n\nPrint results for 20 test data points:")
print("Predicted labels: ", y_pred[20:40])
print("Ground truth    : ", y_test[20:40])

from sklearn.metrics import accuracy_score
print("\n\nAccuracy of 10NN: %.2f %%"
%(100*accuracy_score(y_test, y_pred)))
```

3.4.2. Thực thi thuật toán.

Kết quả sau khi chạy thuật toán:

```
Microsoft Windows [Version 10.0.19041.388]
(c) 2020 Microsoft Corporation. All rights reserved.
(venv) F:\UTC2_University\HK8\DoAnTotNghiep\KNN-divorce>python "kNN-divorce(clear).py"


Print results for 20 test data points:
Predicted labels:  [1 0 0 1 0 0 0 1 1 0 1 1 1 1 1 0 0 1 0 0]
Ground truth    :  [1 0 0 1 1 0 0 1 1 0 1 1 1 1 1 0 0 1 0 0]


Accuracy of 10NN: 96.43 %
```

Hình 3. 12 Kết quả khi thực thi thuật toán

Đối với mô hình này, độ chính xác của bộ thử nghiệm là khoảng 96.43% có nghĩa là đã đưa ra dự đoán đúng cho 96.43% tình trạng hôn nhân trong bộ thử nghiệm. Theo một số giả định toán học, điều này có nghĩa là chúng ta có thể mong đợi mô hình của mình đúng 96.43% đối với tình trạng hôn nhân mới. Đối với ứng dụng này, mức độ chính xác cao này có nghĩa là mô hình có thể đủ tin cậy để sử dụng.

3.5. Xây dựng website demo.

Để minh họa cho thuật toán đã xây dựng, trong phần này tôi sẽ xây dựng một website demo và áp dụng thuật toán đó (được code lại bằng JavaScript) vào trong dự đoán tình trạng hôn nhân. Để làm được điều này tôi đã sử dụng ngôn ngữ HTML, CSS, JavaScript và một số thư viện mã nguồn mở như Bootstrap.

Sau khi tổng hợp và chia 54 đặc điểm trên thành 5 phần chính: Dành thời gian cho nhau, Hiểu nhau, Hòa hợp nhau, Thảo luận, Quan điểm.

3.5.1. Thiết kế thuật toán.

Đầu tiên, cần phải chuẩn bị dữ liệu. Vì thuật toán được code lại toàn bộ bằng JS nên tôi đã chuyển toàn bộ 170 điểm dữ liệu từ file excel thành 170 đối tướng của JS lưu với đuôi file là '.js'.

Sau đó thiết kế thuật toán bằng cách tạo class KNNBrain:

- Hàm khởi tạo:

```
constructor()
{
    this.data = [];

    this.distanceFunction = this.euclidianDistance;
}
```

Hình 3. 13 Phương thức khởi tạo

Khởi tạo biến lưu dữ liệu và lưu khoảng cách.

- Thêm dữ liệu từ file js vào mảng data:

```
addDataset(dataset)
{
    if (!(dataset instanceof Array))
    {
        console.error("dataset needs to be an array of objects");
        return;
    }

    for (let i = 0; i < dataset.length; i++)
    {
        this.data.push(dataset[i]);
    }
}
```

Hình 3. 14 Phương thức addDataset

- Tìm k điểm gần nhất:

```
nearestNeighbours(item, k)
{
    if (k > this.data.length)
    {
        console.error("k is greater than dataset's size");
        return;
    }
    //Mảng lưu k điểm gần nhất
    let nn = [];

    for (let n = 0; n < k; n++)
    {
        let nearest = this.data[0];

        for (let i = 1; i < this.data.length; i++)
        {
            let elt = this.data[i];

            if (this.distanceFunction(elt, item) < this.distanceFunction(nearest, item) && nn.indexOf(elt) < 0)
            {
                nearest = elt;
            }
        }
        //Thêm điểm gần nhất mới không trùng vào mảng
        console.log(nearest);
        nn.push(nearest);
    }

    return nn;
}
```

Hình 3. 15 Phương thức tìm k điểm gần nhất

Trước hết, phải so sánh số k có vượt quá độ dài của tổng các điểm dữ liệu (170) hay không. Nếu k thỏa thì tính khoảng cách tất cả các điểm đến điểm dữ liệu ta xét (điểm dữ liệu mới muốn được dự đoán) và bỏ các điểm gần nhất không trùng vào 1 mảng để lưu lại. Công thức tính khoảng cách là Euclidian.

```
euclidianDistance(a, b)
{
    let sum = 0;

    for (let prop in a.properties)
    {
        if (b.properties.hasOwnProperty(prop))
        {
            let diff = a.properties[prop] - b.properties[prop];
            sum += diff * diff;
        }
    }

    return Math.sqrt(sum);
}
```

Hình 3. 16 Phương thức tính khoảng cách Euclidian

- Phân lớp:

```
classify(item, k = 5)
{   //Mảng k điểm gần nhất
    let nn = this.nearestNeighbours(item, k);

    //Lưu số lượng trùng của mỗi class
    let pickedClasses = {};

    for (let i = 0; i < nn.length; i++)
    {   //Điểm gần nhất thứ i
        let elt = nn[i];

        if (pickedClasses.hasOwnProperty(elt.class))
        {   //Nếu class này đã được thêm thì tăng giá trị số lượng lên
            pickedClasses[elt.class]++;
        }
        else
        {   //Nếu class này chưa được thêm thì thêm vào với số lượng 1
            pickedClasses[elt.class] = 1;
        }
    }

    let bestClass = {
        name: "sample text",
        score: -Infinity
    };

    for (let c in pickedClasses)
    {   //s lần lượt là giá trị số lượng từng class trong k điểm gần nhất
        let s = pickedClasses[c];

        if (s > bestClass.score)
        {   //nếu số lượng lớp đó nhiều hơn score thì gán class đó cho bestClass
            bestClass = {
                name: c,
                score: s
            };
        }
    }
    //trả về class có số lượng nhiều nhất trong k điểm gần nhất
    return bestClass.name;
}
```

Hình 3. 17 Phương thức phân lớp

Duyệt k điểm gần nhất, lưu lại số lượng của từng lớp (nhãn). Sau đó tìm ra nhãn có số lượng nhiều nhất trong k điểm gần nhất.

Thế là xong phần thiết kế class cho thuật toán kNN, kế tiếp ta tạo đối tượng KNNBrain và viết hàm lấy các dữ liệu người dùng nhập vào.

Cuối cùng là dự đoán:

```
function predictDivorce()
{
    // let item =
    createObject(2,2,4,1,0,0,0,0,0,0,1,0,1,1,0,1,0,0,0,1,0,0,0,0,0,0,0,0,0,1,1,2,1,2,0,1,2,1,3,3,2
    //class 1
    // let item = createObject(
    0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,1,

    let item = createObject();

    predictDivorceclass = c.classify(item);

    result.innerHTML = (predictDivorceclass == 0) ? "Chúc mừng! Tình trạng hôn nhân của bạn và "
    + "người ấy vẫn còn tốt, hãy cố gắng vun đắp cho tốt hơn nữa nhé :)"
    : "Lưu ý! Tình trạng hôn nhân của bạn và người ấy khá xấu, có thể dẫn tới ly hôn!";
}
```

Hình 3. 18 Phương thức main

3.5.2. Kiến trúc hệ thống:

- Kiến trúc hạ tầng: Ứng dụng là một website nên chỉ cần 1 thiết bị có thể truy cập web.

- Cấu trúc dữ liệu: Dữ liệu huấn luyện được lưu trực tiếp trên file, không cần sử dụng hệ quản trị cơ sở dữ liệu nào. Thông tin được lấy trực tiếp từ trên web về xử lý.

3.5.3. Giao diện website.

Đường dẫn đến website: http://tinhtranghonnhan.herokuapp.com/

Giao diện trang chủ:

Hình 3. 19 Giao diện trang chủ webdemo

Trang chủ với một số thông tin về tình trạng hôn nhân hiện nay, lí do tạo nên website, biểu mẫu khảo sát dữ liệu và một số đồ thị liên quan.

Giao diện liên hệ:

Hình 3. 20 Giao diện liên hệ webdemo

Đây là thông tin liên hệ bao gồm: Thông tin cá nhân là Tên, Số điện thoại, gmail, địa chỉ; Thời gian làm việc; và các dịch vụ cung cấp.

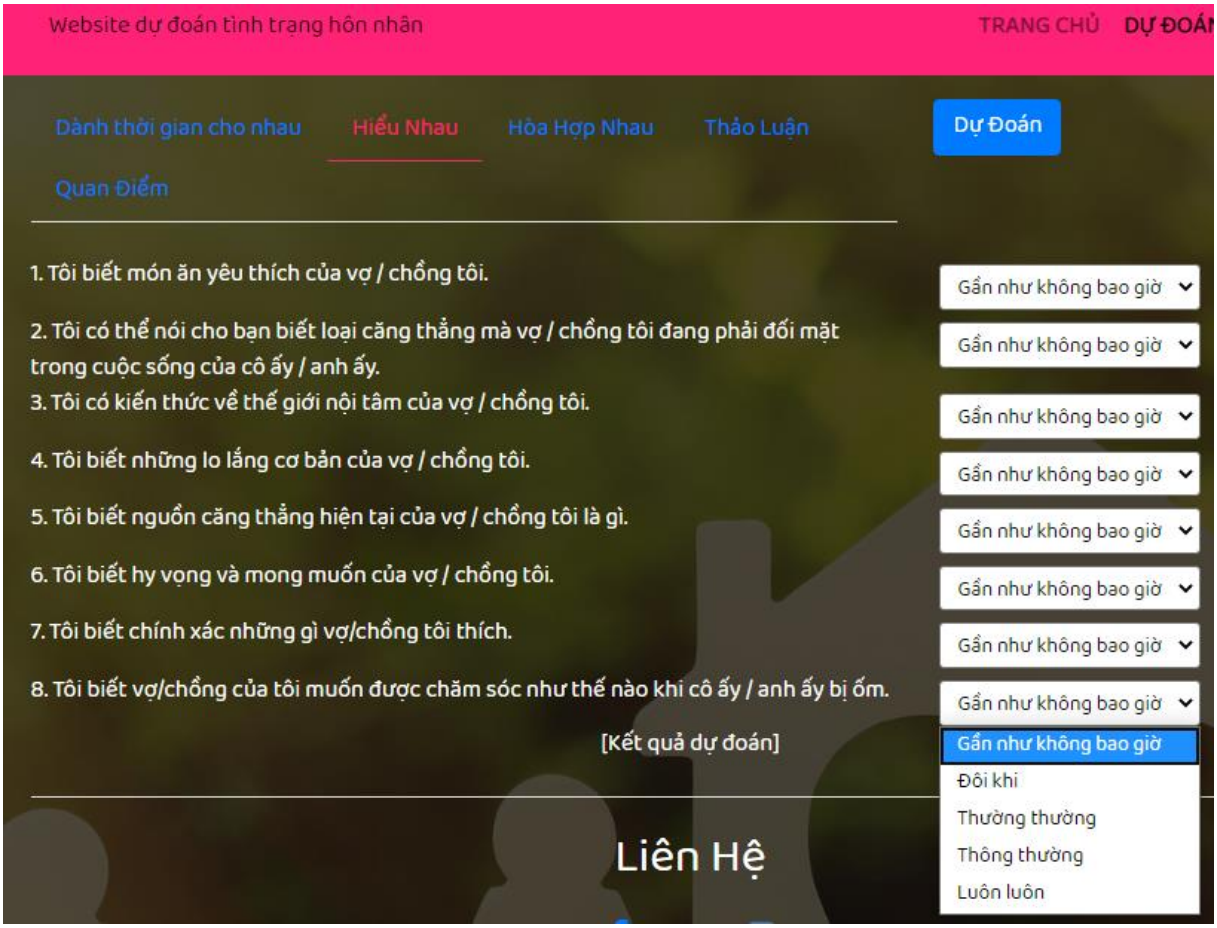
Có thể nhấn vào facebook để theo dõi, nhắn tin hoặc follow youtube, instagram.

Bây giờ đến với chức năng chính của trang web, giao diện trang dự đoán ly hôn:

Hình 3. 21 Giao diện chức năng dự đoán

Trang dự đoán bao gồm 5 tab menu, tổng cổng số câu trong 5 tab tương ứng với 54 đặc điểm trong bộ dữ liệu. Giá trị mặc định cho mỗi câu là gần như không bao giờ, tương ứng với giá trị 0.

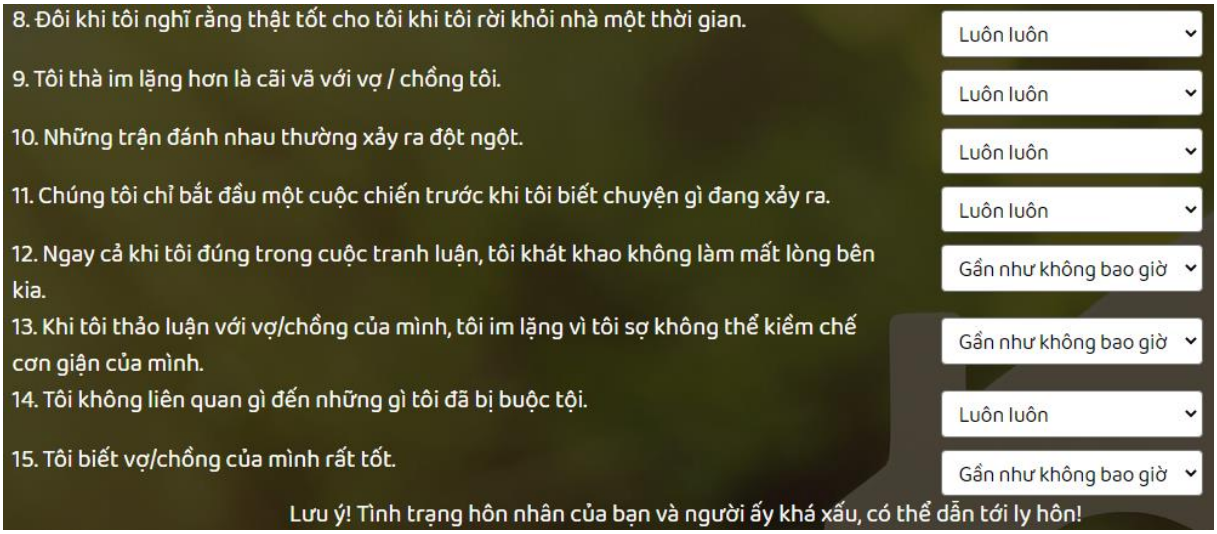
Giao diện khi nhập dữ liệu và chọn dự đoán:



Hình 3. 22 Giao diện khi nhập liệu

Các giá trị từ 0-4 trong dữ liệu được quy đổi theo Thang đo Dự đoán Ly hôn (DPS) trên cơ sở liệu pháp cặp đôi Gottman tương ứng như sau: 0: Gần như không bao giờ, 1: Đôi khi, 2: Thường thường, 3: Thông thường, 4: Luôn luôn.

Khi dự đoán, kết quả sẽ có 2 trường hợp:



Hình 3. 23 Giao diện ra kết quả "ly hôn"

Hình 3. 24 Giao diện ra kết quả "không ly hôn"

3.6. Kết luận chương 4.

Trong chương này, tôi đã trình bày cách ứng dụng thuật toán, để giải bài toán dự đoán tình trạng hôn nhân bằng cách sử dụng mô hình k-nearest neighbors, dựa trên tập dữ liệu mẫu để tiến hành đào tạo, kiểm tra và đánh giá mô hình.

KẾT LUẬN VÀ KIẾN NGHỊ

Kết quả đạt được.

- Biết được các kiến thức nền tảng về Machine Learning – một lĩnh vực đầy tiềm năng trong thời kỳ cách mạng lần thứ 4 (cách mạng công nghiệp 4.0).

- Tìm hiểu được các thuật toán trong Machine Learning, cụ thể là thuật toán K-Nearest Neighbors và xây dựng một mô hình, cài đặt thuật toán KNN.

- Nắm được ý nghĩa của thuật toán, từ đó có thể áp dụng vào các bài toán khác liên quan.

- Xây dựng và ứng dụng được thuật toán vào cài đặt web demo.

Tồn tại.

Bên cạnh những khía cạnh đạt được, do thời gian thực hiện có hạn cùng với kiến thức cũng như trình độ còn nhiều hạn chế nên đã còn những thiếu sót như:

- Tập dữ liệu mẫu được sử dụng còn hạn chế.
- Giao diện chưa đẹp.
- Tính năng chưa đa dạng.

Kiến nghị.

- Dự kiến trong thời gian tới tôi sẽ cố gắng tối ưu thuật toán hơn nữa, thiết kế ứng dụng thân thiện và dễ sử dụng nhất có thể.

- Bên cạnh đó, tôi cũng sẽ cố gắng tìm kiếm, trích xuất, sàng lọc ra những bộ dữ liệu tốt nhất, sát với thực tế ở quốc gia mình nhất để có thể đáp ứng nhu cầu của người tra cứu, tìm kiếm, sử dụng ở nước ta.

-Và dùng các dữ liệu từ 54 đặc điểm [8] để có thể tư vấn cho người dùng nên cải thiện từng đặc điểm như thế nào để hôn nhân bền vững hơn.

- Ngoài ra, tôi sẽ tìm hiểu và nghiên cứu thêm các thuật toán khác để có thể áp dụng vào từng bài toán cụ thể khác nhau.

Vì thời gian thực hiện đề tài có hạn nên trong quá trình làm việc, nghiên cứu không thể tránh khỏi những thiếu sót.

TÀI LIỆU THAM KHẢO

[1]. S hai Shalev-Shwartz, Shai Ben-David, *Understanding Machine Learning: From Theory to Algorithms*, Cambridge University Press, 2014.

[2]. Marcus D. Bloice, Andreas Holzinger, *A Tutorial on Machine Learning and Data Science Tools with Python*, Holzinger Group HCI-KDD, Institute for Medical Informatics, Statistics and Documentation, Medical University of Graz, Graz, Austria.

[3]. Andreas C. Müller, Sarah Guido, *Introduction to Machine Learning with Python*, O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, 2017.

[4]. Daniel T. Larose, Chantal D. Larose, *Discovering Knowledge in Data: An Introduction to Data Mining 2nd Edition*, John Wiley & Sons, Inc., Hoboken, New Jersey, 2005

[5]. Allen B.Downey, *Think Python: How to Think Like a Computer Scientist*, O'Reilly Media, 2012.

[6]. Jon Ducket, *HTML & CSS: Design and Build Websites*, John Wiley & Sons, Inc, 2011.

[7]. Jon Ducket, *JAVASCRIPT & JQUERY: Interactive Front-End Web Development*, John Wiley & Sons, Inc, 2014.

[8]. Mustafa Kemal Yöntem, Kemal ADEM, Tahsin İlhan, Serhat Kılıçarslan, *Divorce Predictors data set Data Set*, 2019, https://archive.ics.uci.edu/ml/datasets/Divorce+Predictors+data+set

[9]. Tất tần tật về Machine Learning & ứng dụng trong những ngành công nghiệp lớn, https://techtalk.vn/tat-tan-tat-moi-kien-thuc-co-ban-ve-machine-learning.html.

[10]. Vũ Hữu Tiệp, *Machine Learning cơ bản*, Vũ Hữu Tiệp's Blog, https://machinelearningcoban.com.

[11]. Cáp Hữu Quân, *Machine Learning là gì?*, Cáp Hữu Quân's Blog, 2017, https://caphuuquan.blogspot.com/2016/05/machine-learning-la-gi.html.

[12]. Hồ Sỹ Hùng, *Machine Learning là gì & tại sao nó lại quan trọng - Phần 1*, techmaster website, 2016, https://techmaster.vn/posts/33836/machine-learning-la-gi-tai-sao-machine-learning-lai-quan-trong.

[13]. Ông Xuân Hồng, *Scikit-learn: K-nearest neighbors*, Fastly, 2015, https://nbviewer.jupyter.org/github/ongxuanhong/hong_notebooks/blob/master/python/knn.ipynb

[14]. *k-nearest neighbors algorithm*, Wikipedia, https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

[15]. Long Nguyen, *Understand Machine Learning*, TheWay2AI blog, 2020, https://theway2ai.com/2020/08/08/understanding-machine-learning/

[16]. Jason Brownlee, *Develop k-Nearest Neighbors in Python From Scratch*, Machine Learning Mastery, 2019, https://machinelearningmastery.com/tutorial-to-implement-k-nearest-neighbors-in-python-from-scratch/?fbclid=IwAR1W-0K06phRFg71VfzNGJNE_SnA46RSDu9Xcsju3N0hZKiVLR-MHP820V4

[17]. Afshine Amidi, Shervine Amidi, *Supervised Learning cheatsheet*, Shervine Amidi's Blog, https://stanford.edu/~shervine/teaching/cs-229/cheatsheet-supervised-learning