

Câu 1

I. Tổng quát

1. Mô hình học máy

- Mô hình học máy là các chương trình máy tính được sử dụng để **nhận dạng** các mẫu trong dữ liệu hoặc đưa ra **dự đoán**.

Machine Learning Model = Model Data + Prediction Algorithm

- Các mô hình học máy được tạo từ các thuật toán học máy, được huấn luyện (training) bằng cách sử dụng dữ liệu được gắn nhãn, không được gắn nhãn hoặc hỗn hợp. Vì vậy, nếu chúng ta huấn luyện một mô hình trên một số dữ liệu huấn luyện và sau đó áp dụng mô hình đó cho dữ liệu mới, thì mô hình đó sẽ có thể suy ra một số mối quan hệ bên trong nó. Mỗi thuật toán học máy khác nhau phù hợp với các mục tiêu sử dụng khác nhau, chẳng hạn như phân loại hoặc dự đoán.
- **Mục tiêu của việc xây dựng mô hình học máy** là giải quyết một vấn đề, mô hình có khả năng phân tích dữ liệu, xác định các mẫu và đưa ra dự đoán hoặc một quyết định.

2. Thuật toán học máy

- Cũng như con người, học là quá trình chuyển hóa tri thức từ việc ghi nhớ những kinh nghiệm trước đó (những bài học trước) thành kiến thức của bản thân để đưa ra quyết định trong tương lai.
- Các thuật toán được sử dụng trong học máy được chia thành **ba** loại: học có giám sát (supervised learning), không giám sát (unsupervised learning) và học tăng cường (reinforcement learning).
 - **Supervised learning** là một hướng tiếp cận của machine learning để làm cho máy tính có khả năng "học". Người ta "huấn luyện" (training) máy tính dựa trên những **quan sát có dán nhãn** (labeled

data). Ý tưởng của supervised learning là: bằng việc ghi nhớ và tổng quát hóa một số quy tắc từ những “kinh nghiệm” có đáp án trước đó, máy tính sẽ có thể trả lời được những câu hỏi dù chưa từng gặp phải, nhưng có mối liên quan.

- Ví dụ ta dạy máy tính " $1 + 1 = 2$ " và hy vọng nó sẽ học được phép tính cộng $x + 1$ và trả lời được là " $2 + 1 = 3$ ". Supervised learning mô phỏng việc con người học bằng cách đưa ra dự đoán của mình cho một câu hỏi, sau đó đối chiếu với đáp án. Sau đó con người rút ra phương pháp để trả lời đúng không chỉ câu hỏi đó, mà cho những câu hỏi có dạng tương tự.
- Trong supervised learning, các dữ liệu **bắt buộc phải được dán nhãn trước**. Mô hình sẽ học cách ánh xạ từ đầu vào sang đầu ra dựa trên dữ liệu huấn luyện. Learning (học) trong mô hình được giám sát đòi hỏi phải tạo ra một hàm có thể được huấn luyện bằng cách sử dụng tập dữ liệu huấn luyện, sau đó áp dụng cho dữ liệu chưa nhìn thấy để đáp ứng một số hiệu suất dự đoán. Mục tiêu là xây dựng hàm sao cho nó khái quát tốt về dữ liệu mà nó chưa từng thấy.
- Supervised learning có thể được chia thành hai loại vấn đề khi khai thác dữ liệu: phân loại và hồi quy:
 - Các bài toán phân loại sử dụng thuật toán để gán chính xác dữ liệu thử nghiệm vào các danh mục cụ thể, chẳng hạn như tách táo khỏi cam. Hoặc trong thế giới thực, thuật toán học có giám sát có thể được sử dụng để phân loại thư rác vào một thư mục riêng biệt với hộp thư đến của bạn. Linear classifiers, support vector machines, decision trees và random forest.
 - Hồi quy là một loại phương pháp học có giám sát khác sử dụng thuật toán để hiểu mối quan hệ giữa các biến phụ thuộc và biến độc lập. Các mô hình hồi quy rất hữu ích trong việc dự đoán các giá trị bằng số dựa trên các điểm dữ liệu khác

nhau, chẳng hạn như dự báo doanh thu bán hàng cho một doanh nghiệp nhất định. Một số thuật toán hồi quy phổ biến là linear regression, logistic regression and polynomial regression.

- **Unsupervised learning** phân tích và phân cụm các tập dữ liệu **không được gán nhãn**. Các thuật toán này khám phá các mẫu ẩn trong dữ liệu mà không cần sự can thiệp của con người (do đó, chúng “không được giám sát”). Mục tiêu là có được thông tin chi tiết từ khối lượng lớn dữ liệu mới. Bản thân máy học sẽ xác định điều gì khác biệt hoặc thú vị so với tập dữ liệu.
- Các mô hình học không giám sát được sử dụng cho ba nhiệm vụ chính: clustering(phân cụm), association(liên kết) và dimensionality reduction(giảm kích thước):
- Clustering là một kỹ thuật khai thác dữ liệu để nhóm dữ liệu không được gán nhãn dựa trên những điểm tương đồng hoặc khác biệt của chúng. Ví dụ: thuật toán phân cụm K-mean gán các điểm dữ liệu tương tự vào các nhóm, trong đó giá trị K biểu thị kích thước của nhóm và mức độ chi tiết. Kỹ thuật này rất hữu ích cho việc phân khúc thị trường, nén hình ảnh, v.v.
- Association là một loại phương pháp học không giám sát khác sử dụng các quy tắc khác nhau để tìm mối quan hệ giữa các biến trong một tập dữ liệu nhất định. Những phương pháp này thường được sử dụng để phân tích giỏ hàng thị trường và các công cụ đề xuất, dọc theo dòng đề xuất “Khách hàng đã mua mặt hàng này cũng đã mua”.
- Dimensionality reduction là một kỹ thuật học được sử dụng khi số lượng tính năng (hoặc kích thước) trong một tập dữ liệu nhất định quá cao. Nó giảm số lượng dữ liệu đầu vào xuống kích thước có thể quản lý được đồng thời duy trì tính toàn vẹn của dữ liệu. Thông thường, kỹ thuật này được sử dụng trong giai đoạn tiền xử lý dữ liệu,

chẳng hạn như khi bộ mã hóa tự động loại bỏ nhiều khối dữ liệu hình ảnh để cải thiện chất lượng hình ảnh.

- **Reinforcement learning** là mô hình (hay còn gọi là “agent”) học từ việc tương tác với một môi trường để đạt được các thưởng và tránh trừng phạt, với khả năng không chỉ học cách ánh xạ đầu vào thành đầu ra mà còn ánh xạ một loạt đầu vào thành đầu ra với các phụ thuộc.
- Học tăng cường tồn tại trong bối cảnh các trạng thái trong môi trường và các hành động có thể thực hiện được ở trạng thái nhất định. Trong quá trình học, thuật toán khám phá ngẫu nhiên các cặp trạng thái - hành động trong một số môi trường (để xây dựng bảng cặp trạng thái - hành động), sau đó trong thực tế thông tin đã học sẽ khai thác phần thưởng của cặp trạng thái - hành động để chọn hành động tốt nhất cho một tình huống nhất định.

3. Tiêu chí đánh giá

- Tiêu chí học trong các mô hình học máy bao gồm accuracy (độ chính xác), interpretability (khả năng diễn giải), complexity (độ phức tạp), training time (thời gian huấn luyện), scalability (khả năng mở rộng), và trade-offs (khả năng đánh đổi).

4. Các loại bài toán

Về cơ bản, học máy cũng tập trung và bốn vấn đề cốt lõi này tương ứng với bốn lớp nhiệm vụ:

- **Phân cụm (Clustering):** Mục tiêu của phân cụm là tìm ra các nhóm dữ liệu có các đặc điểm tương đồng, chẳng hạn như xếp các bức ảnh khuôn mặt gần giống nhau vào các nhóm (nhưng không biết bức ảnh đó của ai và là gì). Đây là khả năng học tập cơ bản nhất của con người - khả năng so sánh sự tương đồng. Dữ liệu được sử dụng trong bài toán phân cụm không được gán nhãn, nhưng có độ tương tự về cấu trúc tự nhiên của dữ liệu.

- **Phân lớp (Classification):** Khác với phân cụm, phân lớp dễ dàng đánh giá hơn. Chúng ta có sẵn các dữ liệu được gán nhãn sẵn thành các lớp, chẳng hạn như: độc hại/không độc hại, gian lận/không gian lận,... Nhiệm vụ của mô hình là quyết định gán nhãn hay phân lớp cho các dữ liệu chưa được gán nhãn. Bài toán phân lớp là một trong những bài toán có ứng dụng rộng rãi nhất trên thực tế, và độ đo đánh giá của nó cũng rất rõ ràng (dựa trên tỷ lệ chính xác, tỷ lệ phân lớp đúng/sai, và một số tiêu chí khác sẽ trình bày trong bài tiếp theo).
- **Hồi quy (Regression):** Đây là bài toán mà đa phần các nhà khoa học vật lý, hóa học,...thực hiện. Đó là việc mô hình hóa và xây dựng các quy tắc tổng quát để mô tả thế giới. Về mặt bản chất, các công thức Newton hay các phương trình nhiệt động,... và tất cả các phương trình được xây dựng trong khoa học đều là bài toán hồi quy. Nghĩa là từ một tập các dữ liệu quan sát được, các nhà khoa học sẽ tìm cách để xây dựng các công thức biểu diễn để mô tả và tổng quát hóa tự nhiên. Các công thức này đều được coi là đúng cho đến khi tồn tại các bộ dữ liệu chứng minh nó sai (vấn đề này hơi lan man sang lĩnh vực triết học một chút). Về mặt bản chất hồi quy là bài toán gán nhãn cho dữ liệu thực, biểu diễn và dự đoán đầu ra dựa trên tổng quát hóa các dữ liệu từ đầu vào để tìm ra một hàm dự đoán. Chẳng hạn xây dựng hàm dự đoán giá nhà, giá cổ phiếu theo thời gian hoặc các biến đầu vào khác. Nếu ta chú ý có thể thấy rằng, thực chất bài toán phân lớp cũng là một trường hợp đặc biệt của bài toán hồi quy, khi giá trị các dự đoán đầu ra thay vì các lớp thì là các giá trị rời rạc đại diện cho lớp. Việc đánh giá một mô hình hồi quy giống như mô hình phân lớp cũng tương đối rõ ràng. Chúng ta có thể đánh giá một mô hình hồi quy thông qua độ lệch dự đoán trung bình của đầu ra. Điểm khó khăn nhất với hồi quy là xác định hàm hồi quy, và vấn đề gây rắc rối nhất là các dữ liệu nhiễu hoặc quan sát dữ liệu sai. Điều này sẽ làm cho các công thức hoàn toàn bị sai hoặc gây hiểu

nhằm. Chẳng hạn tốc độ ánh sáng đã từng bị đo sai và người ta nghĩ rằng vận tốc ánh sáng là không tuyệt đối...

- **Học tăng cường (reinforcement learning):** đây là kiểu bài toán mà máy sẽ được học một hàm mục tiêu sao cho khả năng thích nghi (được nhận phần thưởng nhiều nhất và ít bị phạt nhất) theo môi trường. Chẳng hạn rẽ trái được thưởng 1 điểm còn rẽ phải trừ 1 điểm. Việc học thích nghi theo dữ liệu môi trường nhằm tối ưu hóa một mục tiêu được đặt ra.
- Có nhiều thuật toán khác nhau trong Machine learning, vì vậy việc chọn thuật toán nào phụ thuộc vào độ phù hợp cho tập dữ liệu và mục tiêu của bài toán. Mỗi thuật toán đều có ưu và nhược điểm riêng. Như vậy, tùy trường hợp mà lựa chọn mô hình học máy phù hợp.

II. Phân tích và so sánh các mô hình

1. kNN:

1.1. Giới thiệu về kNN

K-nearest neighbor là một trong những thuật toán supervised-learning trong Machine Learning. KNN được xếp vào loại lazy learning, nó không học một điều gì từ tập dữ liệu học, mọi tính toán được thực hiện khi nó cần dự đoán nhãn của dữ liệu mới. Lớp (nhãn) của một đối tượng dữ liệu mới có thể dự đoán từ các lớp (nhãn) của k hàng xóm gần nó nhất.

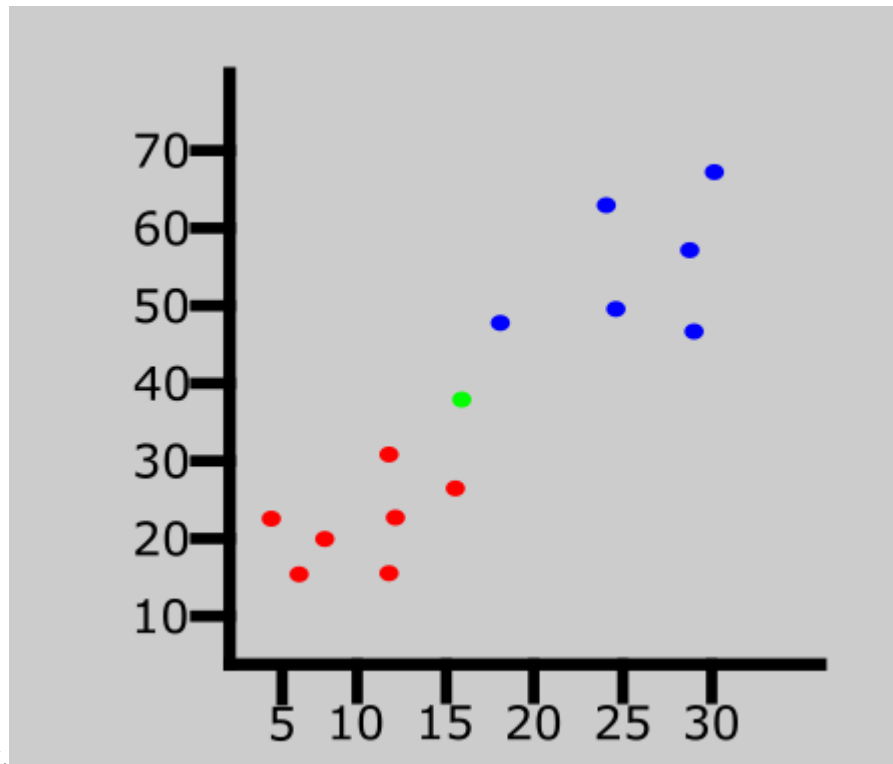
Thuật toán kNN dựa trên những đặc điểm tương đồng, với phương pháp học “lười”, thuật toán KNN rất hữu ích khi bạn đang thực hiện tác vụ nhận dạng mẫu để phân loại đối tượng dựa trên các đặc điểm khác nhau.

1.2. Thuật toán hoạt động

- Vì là thuật toán “lười” nên kNN trong quá trình training nó không “tổng quát hóa kiến thức” (learning). Nói cách khác, không có giai đoạn training rõ ràng và điều đó cũng có nghĩa giai đoạn training diễn ra rất nhanh.

- Thiếu tính khái quát có nghĩa là kNN đưa ra quyết định dựa trên toàn bộ tập dữ liệu training.
- **kNN trong bài toán phân loại:** nhãn dự đoán được xác định bằng cách dựa vào k láng giềng gần nhất, nghĩa là lớp nhãn đa số trong tập hợp k sẽ được trả về.
 - Bước 1: Chọn số **k** của hàng xóm
 - Bước 2: Tính khoảng cách Euclide của k số hàng xóm
 - Bước 3: Lấy k hàng xóm gần nhất theo khoảng cách Euclide được tính toán.
 - Bước 4: Trong số k hàng xóm này, đếm số lượng điểm dữ liệu trong mỗi danh mục.
 - Bước 5: Gán các điểm dữ liệu mới cho danh mục đó mà số lượng điểm lân cận là nhiều nhất.
- Ví dụ:

- Ta có 2 loại nhãn là Red và Blue, điểm xanh lá đại diện cho điểm dữ liệu mới thêm vào tập dữ liệu. Giờ ta sẽ tiến hành phân loại của điểm



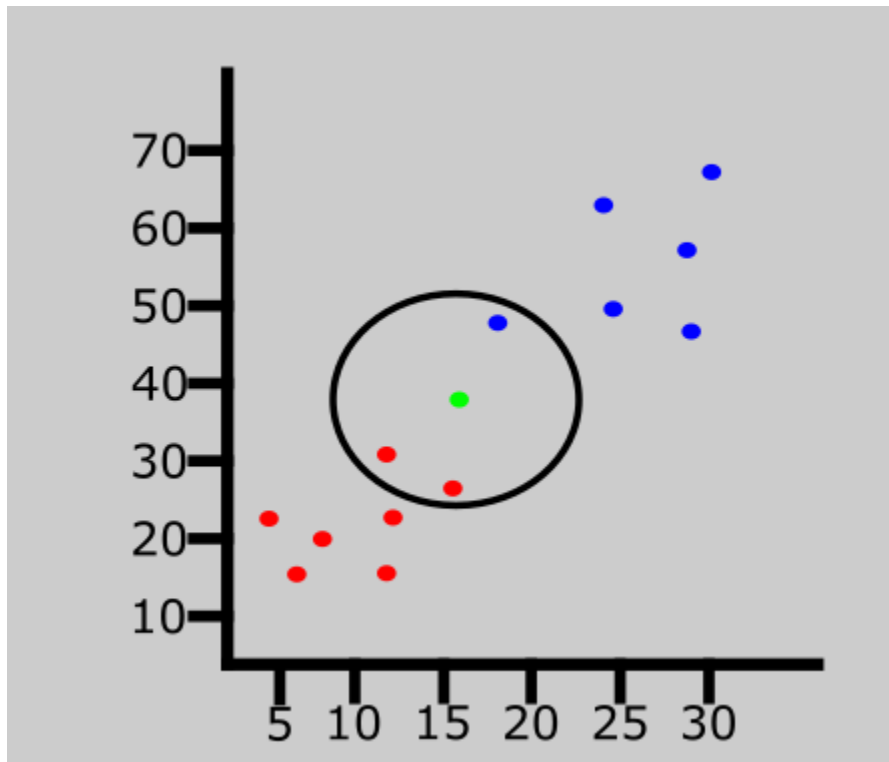
này.

- Chọn k biểu thị số lượng hàng xóm cần xem xét trước khi tiến hành phân loại. Giả sử $k = 3$.
- Tiếp theo, khoảng cách của từng phần tử sẽ được tính theo khoảng cách Euclide theo công thức:

$$disc(d) = \sqrt{(x - a)^2 + (y - b)^2}$$

- Dựa theo kết quả khoảng cách đã tính ở trên, ta chọn được k phần tử vào phân loại

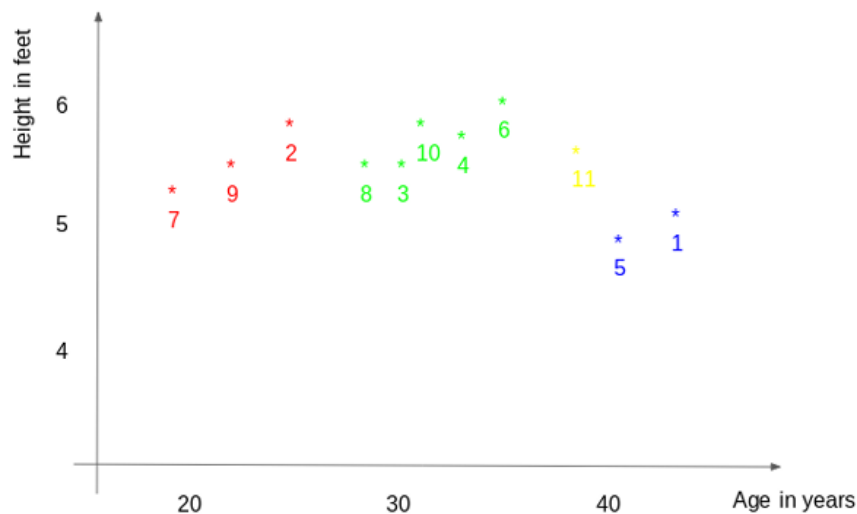
- Trong k hàng xóm này, ta thấy được nhãn chiếm đa số là Red nên điểm dữ liệu mới được gán nhãn Red



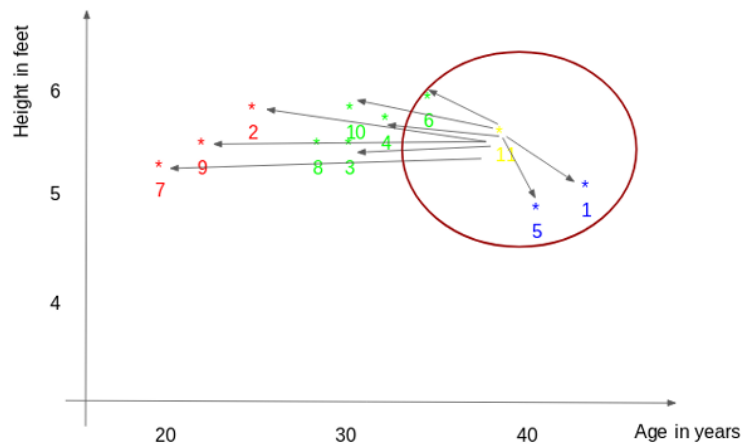
- **kNN trong bài toán hồi quy:** giá trị trung bình của các giá trị hàm lân cận gần nhất được trả về dưới dạng giá trị dự đoán.
 - Ví dụ, ta có tập dữ liệu như sau:

ID	Height	Age	Weight
1	5	45	77
2	5.11	26	47
3	5.6	30	55
4	5.9	34	59
5	4.8	40	72
6	5.8	36	60
7	5.3	19	40
8	5.8	28	60
9	5.5	23	45
10	5.6	32	58
11	5.5	38	?

- Tập dữ liệu được biểu diễn dưới dạng biểu đồ



- Trực y biểu thị chiều cao của một người (tính bằng feet) và trục x biểu thị tuổi (tính bằng năm). Các điểm được đánh số theo giá trị ID. Điểm màu vàng (ID 11) là điểm cần kiểm tra.
- Thuật toán như sau:
 - Tính khoảng cách của từng đối tượng theo công thức đã nêu trên.
 - Dựa theo kết quả khoảng cách đã tính ở trên, ta chọn được k đối



tượng.

- Giá trị trung bình của các điểm dữ liệu này là dự đoán cuối cùng cho điểm mới. Ở đây, chúng ta có trọng lượng ID11 = $(77+72+60)/3 = 69,66$ kg.

1.3. Bài toán

- kNN có thể được dùng trong cả hai bài toán là dự đoán “hồi quy” (regression) và dự đoán “phân loại”(classification).
- Tuy nhiên, nó chủ yếu được sử dụng trong classification (phân loại) vì nó dựa trên tất cả các tham số được đánh giá khi xác định khả năng sử dụng của một kỹ thuật dựa trên: khả năng dự đoán, thời gian tính toán và dễ dàng giải thích đầu ra.
- Thuật toán KNN phù hợp với tất cả các thông số cần cân nhắc. Nhưng chủ yếu, nó được sử dụng do tính dễ hiểu và thời gian tính toán thấp.
- Tập dữ liệu lý tưởng là khi có nhiều loại dữ liệu khác nhau và đều đã được gán “nhãn” (label).

1.4. Ưu điểm

- Thời gian tính toán nhanh
- Thuật toán đơn giản – dễ giải thích
- Đa năng - hữu ích cho hồi quy và phân loại
- Không có giả định về dữ liệu – không cần đưa ra các giả định bổ sung, điều chỉnh một số tham số hoặc xây dựng mô hình. Điều này làm cho nó trở nên thích hợp trong trường hợp dữ liệu dưới dạng phi tuyến tính.

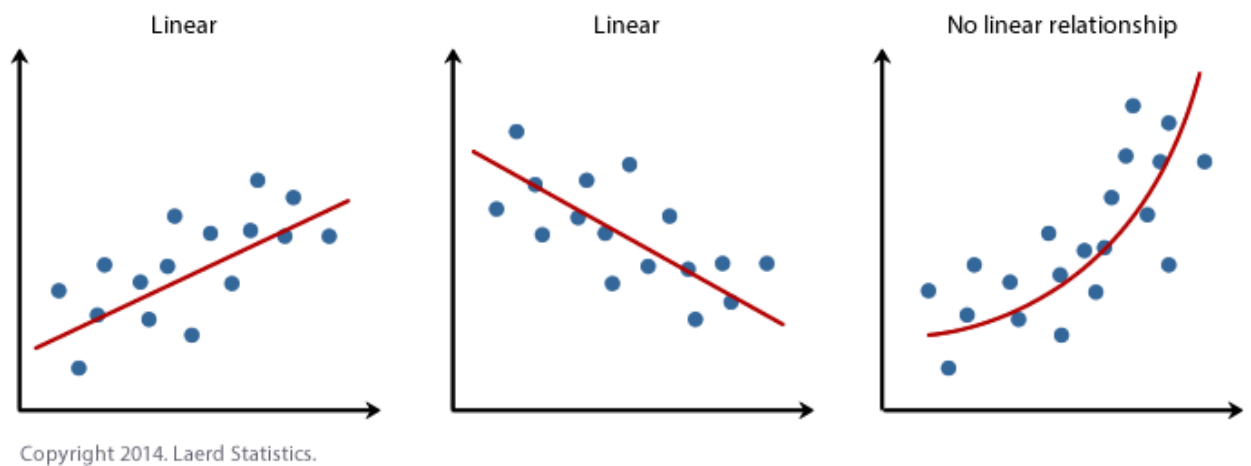
1.5. Nhược điểm

- Chọn giá trị k thích hợp
- K-NN có thể rất dễ triển khai nhưng khi tập dữ liệu tăng thì hiệu quả hoặc tốc độ thuật toán giảm rất nhanh
- Độ chính xác phụ thuộc vào chất lượng dữ liệu
- Nhạy cảm với quy mô của dữ liệu và các tính năng không liên quan
- Yêu cầu bộ nhớ cao – cần lưu trữ tất cả dữ liệu huấn luyện (data training)
- Vì nó lưu trữ tất cả các “training” nên nó có thể tốn kém về mặt tính toán

2. Linear Regression

2.1. Giới thiệu về Linear Regression

- Linear hay tuyến tính hiểu một cách đơn giản là thẳng, phẳng. Trong không gian hai chiều, một hàm số được gọi là tuyến tính nếu đồ thị của nó có dạng một đường thẳng. Trong không gian ba chiều, một hàm số được gọi là tuyến tính nếu đồ thị của nó có dạng một mặt phẳng. Trong không gian nhiều hơn 3 chiều, khái niệm mặt phẳng không còn phù hợp nữa, thay vào đó, một khái niệm khác ra đời được gọi là siêu mặt phẳng (hyperplane). Các hàm số tuyến tính là các hàm đơn giản nhất, vì chúng thuận tiện trong việc hình dung và tính toán.

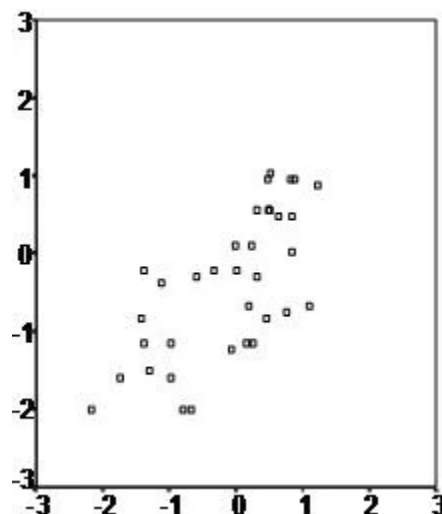


- Hồi quy tuyến tính được sử dụng để dự đoán các kết quả đầu ra liên tục trong đó có mối quan hệ tuyến tính giữa các đặc điểm của tập dữ liệu và biến đầu ra. Nó được sử dụng cho các bài toán hồi quy trong đó bạn đang cố gắng dự đoán điều gì đó với vô số câu trả lời khả dĩ, được sử dụng rộng rãi trong nhiều lĩnh vực khác nhau cho các nhiệm vụ như dự báo, phân tích xu hướng và khám phá mối tương quan chẳng hạn như giá một ngôi nhà dựa vào vị trí địa lý.
- Chức năng của hồi quy tuyến tính** là mô hình hóa và phân tích mối quan hệ giữa một biến phụ thuộc (mục tiêu) và một hoặc nhiều biến độc lập (đặc điểm). Nó giúp chúng ta hiểu các biến độc lập ảnh hưởng như thế nào đến biến phụ thuộc và cho phép chúng ta đưa ra dự đoán dựa trên mối quan hệ đã học. Hồi quy tuyến tính ước tính đường phù hợp nhất thể hiện mối quan hệ này, cho phép chúng tôi phân tích và đưa ra dự đoán cho các điểm dữ liệu mới.

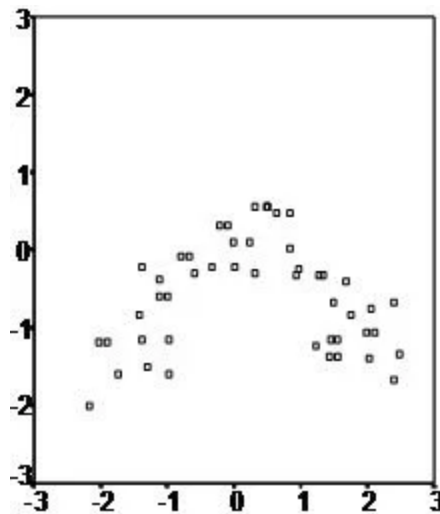
- Có 2 loại Linear Regression
 - a. Simple Linear Regression và Multiple Linear Regression
 - b. Non-Linear Regression: khi đường thẳng được biểu diễn dưới dạng đường cong.

2.2. Thuật toán hoạt động

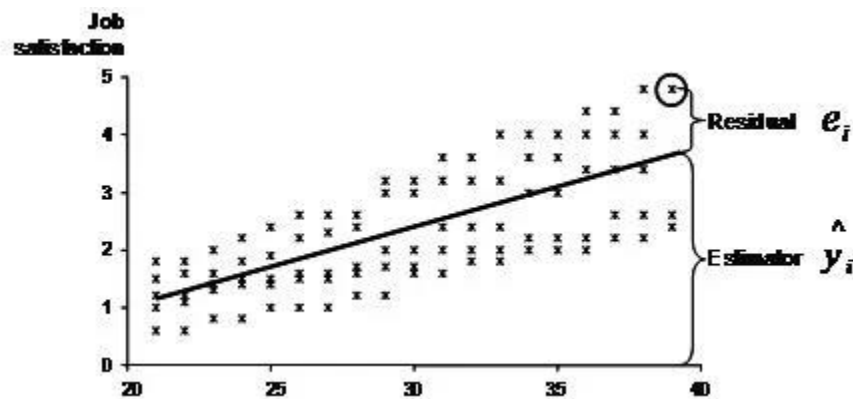
- Đường hồi quy có thể là mối quan hệ tuyến tính dương hoặc mối quan hệ tuyến tính âm:
 - a. Positive Linear Relationship - Mối quan hệ tuyến tính dương: Khi biến độc lập và biến phụ thuộc tương quan đồng biến (tức là cùng tăng hoặc cùng giảm)
 - b. Negative Linear Relationship - Mối quan hệ tuyến tính âm: Khi biến độc lập và biến phụ thuộc tương quan nghịch biến (tức là biến này tăng thì biến kia giảm và ngược lại).
- Nó bao gồm 3 giai đoạn:
 - a. Phân tích mối tương quan và định hướng của dữ liệu
 - Biểu đồ phân tán đầu tiên cho thấy mối quan hệ tuyến tính dương giữa hai biến. Dữ liệu phù hợp để chạy phân tích hồi quy.



- Biểu đồ phân tán thứ hai dường như có hình chữ U nghịch đảo, điều này cho thấy đường hồi quy có thể không phải là cách tốt nhất để giải thích dữ liệu.

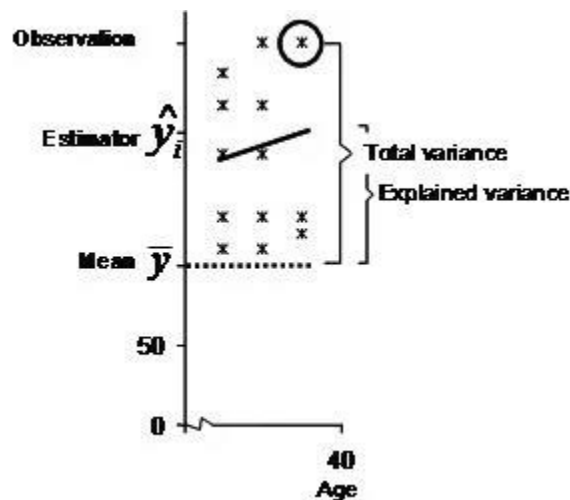


- Sau bước đầu tiên ta đã tiến hành xây dựng mô hình, tức là biến X đó có ảnh hưởng nhân quả đến biến Y và mối quan hệ của chúng là tuyến tính.
- b. Ước tính mô hình, tức là điều chỉnh đường thẳng
 - Ví dụ trong việc mô hình hóa mối quan hệ giữa tuổi tác và sự hài lòng trong công việc.



- Khi chúng ta vẽ một đường thẳng qua biểu đồ phân tán, đường hồi quy biểu thị mức độ hài lòng công việc ước tính cho một độ tuổi nhất định. Bằng phương pháp bình phương tối thiểu ta thu được biểu thức:
- $$\sum e_i^2 = \sum (y_i - \hat{y}_i)^2 = \sum (y_i - b_0 - b_1 x_i)^2 \rightarrow \min \Rightarrow \hat{y}_i = b_0 + b_1 x_i$$
- Ví dụ, kết quả của phương trình này sẽ là $y_i = 1 + 0,1 * x_i$. Điều này có nghĩa là cứ mỗi năm tuổi chúng ta kỳ vọng mức độ hài lòng trong công việc sẽ tăng thêm 0,1.

- Thước đo chính cho tính hợp lệ của đường tuyến tính ước tính là R^2 .
 $R^2 = \text{tổng phương sai} / \text{phương sai được giải thích}$. Biểu đồ sau minh họa các khái niệm chính để tính R^2 . Trong ví dụ trên, R^2 xấp xỉ 0,6, điều này có nghĩa là 60% tổng phương sai được giải thích bằng mối quan hệ giữa độ tuổi và sự hài lòng.



- Ta có thể dễ dàng thấy số lượng dữ liệu đầu vào và số lượng biến độc lập sẽ làm tăng R^2 . Tuy nhiên, việc overfitting xảy ra khi mô hình không còn hiệu quả nữa. Để xác định xem mô hình có được trang bị hiệu quả hay không, R^2 đã hiệu chỉnh sẽ được tính toán, được xác định:

$$R^2_c = R^2 - \frac{J(1-R^2)}{N-J-1}$$

- J là số lượng biến độc lập và N là kích thước tập mẫu.
- Như bạn có thể thấy, tập mẫu càng lớn thì tác động của một biến độc lập bổ sung trong mô hình càng nhỏ. Trong ví dụ này, $R^2_c = 0,6 - 1(1 - 0,6)/95 - 1 - 1 = 0,5957$. Do đó, mô hình khá phù hợp khi chỉ có một biến độc lập trong phân tích.

c. Đánh giá tính hợp lệ và hữu ích của mô hình

- Hồi quy tuyến tính sử dụng hai thử nghiệm để kiểm tra xem mô hình tìm thấy và các hệ số ước tính có thể được tìm thấy trong tổng thể chung mà mẫu được rút ra hay không.

- Đầu tiên, F-test kiểm tra mô hình tổng thể. Giả thuyết không là các biến độc lập không có ảnh hưởng đến biến phụ thuộc. Nói cách khác, kiểm định F của hồi quy tuyến tính kiểm tra xem $R^2 = 0$ hay không.
- Thứ hai, nhiều thử nghiệm t phân tích tầm quan trọng của từng hệ số và điểm chặn. Kiểm định t có giả thuyết không rằng hệ số/điểm chặn bằng 0.

2.3. Bài toán

- Hồi quy tuyến tính phù hợp với các tập dữ liệu có **mối quan hệ tuyến tính** giữa các dữ liệu cung cấp và biến đầu ra.

- Đánh giá mô hình:

a. Cost Function

- Kết quả thuật toán thu được hoặc dự đoán được gọi là \hat{y}
- Sự khác biệt giữa giá trị thực tế và giá trị dự đoán là sai số, tức là $y - \hat{y}$.
- Các giá trị của $y - \hat{y}$ (hàm mất mát) thu được khác nhau khi mô hình liên tục cố gắng tìm ra mối quan hệ tốt nhất. Tổng trung bình của tất cả các giá trị hàm mất mát được gọi là “hàm chi phí” (cost function). Thuật toán cố gắng đạt được giá trị tối thiểu của hàm chi phí.

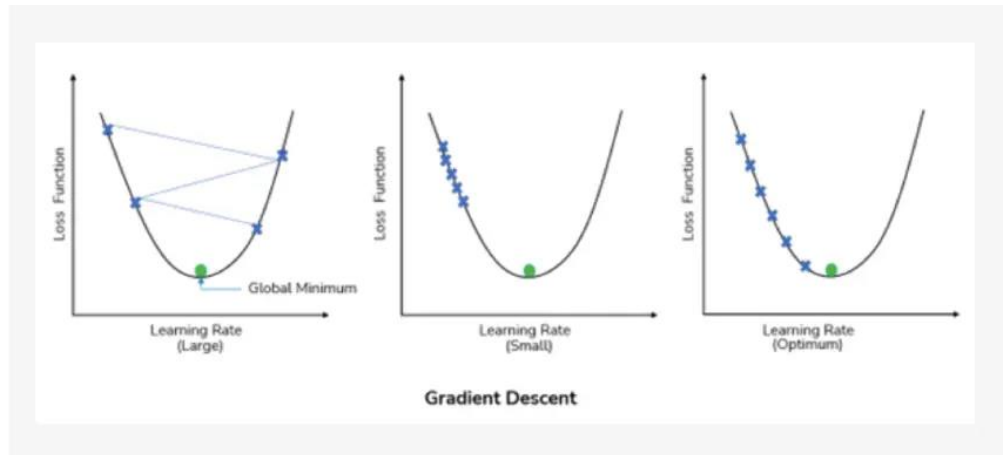
$$\text{minimize } \frac{1}{n} \sum_{i=1}^n (\text{pred}_i - y_i)^2$$

$$J = \frac{1}{n} \sum_{i=1}^n (\text{pred}_i - y_i)^2$$

- Trong đó, J = hàm chi phí, n = số lượng quan sát ($i = 1$ đến n), \sum = tổng, pred_i = đầu ra dự đoán và y_i = giá trị thực tế.

b. Gradient Descent

- Đó là một phương pháp tối ưu hóa phổ biến được sử dụng trong các mô hình học máy đào tạo bằng cách giảm sai sót giữa kết quả thực tế và kết quả dự đoán. Mục tiêu chính của việc giảm độ dốc là giảm thiểu hàm lỗi bằng cách lặp tham số.



- Tốc độ học chậm hơn giúp đạt được mức tối thiểu toàn cầu nhưng mất thời gian dài bất thường và tốn kém về mặt tính toán. Tốc độ học nhanh hơn có thể khiến mô hình đi chệch hướng và dẫn đến vị trí không mong muốn, gây khó khăn cho việc quay lại đúng hướng để đạt mức tối thiểu toàn cục. Do đó, tốc độ học không được quá chậm cũng không quá nhanh nếu muốn đạt được mức tối thiểu toàn cục một cách hiệu quả.

2.4. Ưu điểm

- Hồi quy tuyến tính dễ thực hiện và dễ diễn giải các hệ số đầu ra hơn.
- Khi bạn biết mối quan hệ giữa biến độc lập và biến phụ thuộc có mối quan hệ tuyến tính thì thuật toán này là tốt nhất để sử dụng vì nó ít phức tạp hơn so với các thuật toán khác.
- Hồi quy tuyến tính dễ bị overfitting nhưng có thể tránh được bằng cách sử dụng một số kỹ thuật giảm kích thước, kỹ thuật chính quy hóa (L1 và L2) và xác thực chéo.

2.5. Nhược điểm

- Hạn chế đầu tiên của Linear Regression là nó rất nhạy cảm với nhiễu (sensitive to noise). Vì vậy, trước khi thực hiện Linear Regression, ta cần phải làm sạch (loại bỏ nhiễu) dữ liệu, gọi là tiền xử lý (pre-processing).
- Linear Regression là nó không biểu diễn được các mô hình phức tạp.
- Hồi quy tuyến tính là một công cụ tuyệt vời để phân tích mối quan hệ giữa các biến nhưng nó không được khuyến khích cho hầu hết các ứng dụng thực tế vì nó đơn giản hóa quá mức các vấn đề trong thế giới thực bằng cách giả định mối quan hệ tuyến tính giữa các biến.

3. Naive Bayes classifiers

3.1. Giới thiệu

- Đây là một kỹ thuật phân loại dựa trên Định lý Bayes, Naive Bayes classifiers giả định rằng sự hiện diện của một thuộc tính cụ thể trong một lớp không liên quan đến sự hiện diện của bất kỳ thuộc tính nào khác.
- Naïve Bayes classifiers là một thuật toán học máy có giám sát phổ biến được sử dụng cho các **tác vụ phân loại** như phân loại văn bản. Nó mô hình hóa việc phân phối đầu vào cho một lớp hoặc danh mục nhất định. Cách tiếp cận này dựa trên giả định rằng các tính năng của dữ liệu đầu vào độc lập có điều kiện đối với lớp, cho phép thuật toán đưa ra dự đoán nhanh chóng và chính xác.

3.2. Thuật toán hoạt động

- Bên dưới là tập dữ liệu huấn luyện về thời tiết và biến mục tiêu tương ứng 'Play'. Bây giờ, chúng ta cần phân loại người chơi có chơi hay không dựa trên điều kiện thời tiết.
 - a. Bước 1: Chuyển tập dữ liệu thành bảng tần số
 - b. Bước 2: Tạo bảng Likelihood bằng cách tìm xác suất

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Likelihood table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
All	5	9
	=5/14	=9/14
	0.36	0.64

c. Bước 3: Sử dụng phương trình Naive Bayesian để tính xác suất hậu nghiệm. Lớp có xác suất hậu nghiệm cao nhất là kết quả của dự đoán.

- **Vấn đề:** Người chơi sẽ thi đấu nếu trời nắng. Tuyên bố này có đúng không?
- Chúng ta có thể giải quyết nó bằng phương pháp xác suất hậu nghiệm đã thảo luận ở trên.
- $P(\text{Có} \mid \text{Nắng}) = P(\text{Nắng} \mid \text{Có}) * P(\text{Có}) / P(\text{Nắng})$
- Ở đây $P(\text{Sunny} \mid \text{Yes}) * P(\text{Yes})$ nằm ở tử số và $P(\text{Sunny})$ nằm ở mẫu số.
- Ở đây chúng ta có $P(\text{Nắng} \mid \text{Có}) = 3/9 = 0,33$, $P(\text{Nắng}) = 5/14 = 0,36$, $P(\text{Có}) = 9/14 = 0,64$
- Bây giờ, $P(\text{Có} \mid \text{Nắng}) = 0,33 * 0,64 / 0,36 = 0,60$, có xác suất cao hơn.
- Naive Bayes sử dụng một phương pháp tương tự để dự đoán xác suất của các lớp khác nhau dựa trên các thuộc tính khác nhau.

3.3. Bài toán

- Thuật toán này chủ yếu được sử dụng trong phân loại văn bản (nlp) và với các vấn đề có nhiều lớp.
- Trình phân loại Naive Bayesian là một trình phân loại ham học hỏi và nó có tốc độ siêu nhanh. Vì vậy, nó có thể được sử dụng để đưa ra dự đoán trong thời gian thực.
- Dự đoán nhiều lớp: Thuật toán này cũng nổi tiếng với tính năng dự đoán nhiều lớp. Ở đây chúng ta có thể dự đoán xác suất của nhiều loại biến mục tiêu.
- Phân loại văn bản/ Lọc thư rác/ Phân tích cảm xúc: Trình phân loại Naive Bayesian chủ yếu được sử dụng trong phân loại văn bản (do kết quả tốt hơn

trong các vấn đề đa lớp và quy tắc độc lập) có tỷ lệ thành công cao hơn so với các thuật toán khác. Do đó, nó được sử dụng rộng rãi trong lọc Thư rác (xác định e-mail thư rác) và Phân tích tình cảm (trong phân tích mạng xã hội, để xác định tình cảm tích cực và tiêu cực của khách hàng)

- Hệ thống đề xuất: Bộ phân loại Naive Bayes và Bộ lọc cộng tác cùng nhau xây dựng Hệ thống đề xuất sử dụng kỹ thuật học máy và khai thác dữ liệu để lọc thông tin không nhìn thấy và dự đoán liệu người dùng có muốn một tài nguyên nhất định hay không.

3.4. Ưu điểm

- Ít phức tạp: So với các thuật toán phân loại khác, Naive Bayes được coi là bộ phân loại đơn giản hơn vì các tham số dễ ước tính hơn.
- Chia tỷ lệ tốt: So với hồi quy logistic, Naive Bayes được coi là một công cụ phân loại nhanh và hiệu quả, khá chính xác khi đảm bảo giả định độc lập có điều kiện. Nó cũng có yêu cầu lưu trữ thấp.
- Có thể xử lý dữ liệu nhiều chiều: Các trường hợp sử dụng, chẳng hạn như phân loại tài liệu, có thể có số lượng kích thước cao, điều này có thể khó quản lý đối với các trình phân loại khác.

3.5. Nhược điểm

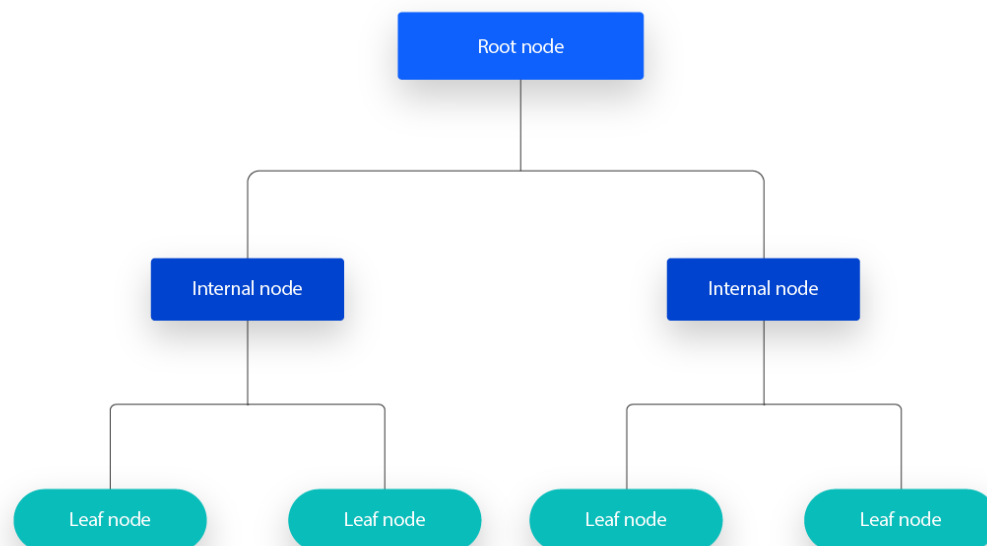
- Tuân theo tần số 0: Tần số 0 xảy ra khi biến phân loại không tồn tại trong tập huấn luyện. Ví dụ: hãy tưởng tượng rằng chúng ta đang cố gắng tìm công cụ ước tính khả năng tôi đã cho từ “thưa ngài” cho lớp “thư rác”, nhưng từ “thưa ngài” không tồn tại trong dữ liệu đào tạo. Xác suất trong trường hợp này sẽ bằng 0 và vì bộ phân loại này nhân tất cả các xác suất có điều kiện với nhau, điều này cũng có nghĩa là xác suất hậu nghiệm sẽ bằng 0. Để tránh vấn đề này, việc làm mịn laplace có thể được tận dụng (*làm mịn laplace là một kỹ thuật làm mịn dữ liệu dạng phân loại (categorical data), một giá trị nhỏ gọi là pseudo-count sẽ được thêm vào để làm thay đổi xác suất đầu ra*).

- Giả định cốt lõi không thực tế: Mặc dù giả định độc lập có điều kiện nhìn chung hoạt động tốt nhưng giả định này không phải lúc nào cũng đúng, dẫn đến việc phân loại không chính xác.

4. Decision Tree

4.1. Giới thiệu

- Cây quyết định là một thuật toán học có giám sát phi tham số. Cây quyết định có thể được sử dụng trong các bài toán phân loại hoặc hồi quy và rất hữu ích cho các **bộ dữ liệu phức tạp**. Nó có cấu trúc cây phân cấp, bao gồm nút gốc, nhánh, nút bên trong và nút lá.



- Chúng hoạt động bằng cách chia tập dữ liệu, theo cấu trúc dạng cây, thành các tập con ngày càng nhỏ hơn, sau đó dựa trên tập dữ liệu huấn luyện đưa ra dự đoán mà ví dụ mới sẽ rơi vào.
- Trong giai đoạn training (huấn luyện), từ dữ liệu thuật toán sẽ học ra model, model có thể dự đoán giá trị của “target variable” (biến mục tiêu) dựa trên tập dữ liệu huấn luyện.
- Khi dữ liệu có hình dạng phi tuyến tính, ta có thể sử dụng cây quyết định để thực hiện công việc nắm bắt tính phi tuyến tính trong dữ liệu tốt hơn bằng cách chia không gian thành các không gian con nhỏ hơn tùy vào từng yêu cầu.

4.2. Thuật toán hoạt động

- Trong cây quyết định, để dự đoán lớp của tập dữ liệu đã cho, thuật toán bắt đầu từ node **root** của cây. Thuật toán này so sánh các giá trị của thuộc tính root với thuộc tính ghi lại (tập dữ liệu thực) và dựa trên so sánh, đi theo nhánh và nhảy đến node tiếp theo.
- Ở node tiếp theo, thuật toán lại so sánh giá trị thuộc tính với các node phụ khác và tiến xa hơn. Nó tiếp tục quá trình cho đến khi đến node lá của cây. Cụ thể theo các bước sau:
 - a. Bước 1: Bắt đầu cây với node **root**, gọi là S, chứa tập dữ liệu hoàn chỉnh.
 - b. Bước 2: Tìm thuộc tính tốt nhất trong tập dữ liệu bằng cách sử dụng **Thuốc đo lựa chọn thuộc tính (ASM)** (*Information gain là một trong những kỹ thuật dùng trong ASM*).
 - c. Bước 3: Chia S thành các tập con chứa các giá trị có thể có của các thuộc tính tốt nhất.
 - d. Bước 4: Tạo node cây quyết định chứa thuộc tính tốt nhất.
 - e. Bước 5: Tạo đệ quy các cây quyết định mới bằng cách sử dụng các tập hợp con của tập dữ liệu được tạo ở bước 3. Tiếp tục quá trình này cho đến khi đạt đến giai đoạn mà bạn không thể phân loại thêm các node và gọi node cuối cùng là node lá.

4.3. Đánh giá mô hình






















- Các chỉ số để đánh giá mô hình bao gồm: entropy, information gain. Chúng giúp đánh giá chất lượng của từng điều kiện thử nghiệm và khả năng phân loại mẫu thành một lớp tốt như thế nào.
- **Entropy - độ hỗn loạn**
 - Entropy là một khái niệm bắt nguồn từ lý thuyết thông tin, đo lường mức độ tạp chất của các giá trị mẫu. Nó được xác định bằng công thức sau, trong đó:

$$\text{Entropy}(S) = - \sum_{c \in C} p(c) \log_2 p(c)$$

- S đại diện cho tập dữ liệu entropy được tính toán
- c đại diện cho các lớp trong tập hợp, S
- $p(c)$ biểu thị tỷ lệ các điểm dữ liệu thuộc lớp c trên tổng số điểm dữ liệu trong tập hợp, S
- Giá trị entropy có thể nằm trong khoảng từ 0 đến 1.
 - Nếu tất cả các mẫu trong tập dữ liệu S thuộc về một lớp thì entropy sẽ bằng 0.
 - Nếu một nửa số mẫu được phân loại vào một lớp và nửa còn lại thuộc lớp khác thì entropy sẽ đạt mức cao nhất là 1.
 - Để chọn được đặc trưng tốt nhất để phân tách và tìm ra cây quyết định tối ưu, thuộc tính có giá trị nhỏ nhất lượng entropy nên được sử dụng.
- **Information gain**
 - Information gain thể hiện sự khác biệt về entropy trước và sau khi phân chia trên một thuộc tính nhất định. Thuộc tính có information gain cao nhất sẽ tạo ra sự phân chia tốt nhất vì nó thực hiện công việc phân loại dữ liệu huấn luyện tốt nhất theo phân loại mục tiêu của nó. Information gain thường được biểu diễn bằng công thức sau, trong đó:

$$\text{Information Gain}(S,a) = \text{Entropy}(S) - \sum_{v \in \text{values}(a)} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

- a đại diện cho một thuộc tính hoặc nhãn lớp cụ thể
- $\text{Entropy}(S)$ là entropy của tập dữ liệu, S
- $|S_v|/|S|$ biểu thị tỷ lệ của các giá trị trong S_v với số giá trị trong tập dữ liệu, S
- $\text{Entropy}(S_v)$ là entropy của tập dữ liệu, S_v
- Ví dụ, ta có tập dữ liệu như sau:

Day	Outlook	Temp	Humidity	Wind	Tennis
1	 Sunny	Hot	 High	 Weak	No
2	 Sunny	Hot	 High	 Strong	No
3	 Overcast	Hot	 High	 Weak	Yes
4	 Rain	Mild	 High	 Weak	Yes
5	 Rain	Cool	 Normal	 Weak	Yes
6	 Rain	Cool	 Normal	 Strong	No
7	 Overcast	Cool	 Normal	 Weak	Yes

- Entropy của tập dữ liệu:
 - $p_{yes} = -\left(\frac{9}{14}\right) * \log_2\left(\frac{9}{14}\right) = 0.41$

$$\blacksquare p_{no} = -\left(\frac{5}{14}\right) * \log_2\left(\frac{5}{14}\right) = 0.53$$

$$\blacksquare Entropy(tennis) = 0.41 + 0.53 = 0.94$$

- Sau đó chúng ta có thể tính toán information gain cho từng thuộc tính riêng lẻ.

Ví dụ: information gain của “Humidity” sẽ như sau:

$$Gain(Tennis, Humidity) = (0.94) - \left(\frac{7}{14}\right) * (0.985) - \left(\frac{7}{14}\right) * (0.592) = 0.151$$

- $\frac{7}{14}$ biểu thị tỷ lệ các giá trị trong đó độ ẩm ở mức “high” trên tổng số giá trị độ ẩm. Trong trường hợp này, số giá trị trong đó độ ẩm bằng “high” cũng giống như số giá trị trong đó độ ẩm bằng “normal”.
- **0,985** là entropy khi Humidity = “high”
- **0,59** là entropy khi Humidity = “normal”
- Sau đó, lặp lại tính toán về mức tăng thông tin cho từng thuộc tính trong bảng trên và chọn thuộc tính có mức tăng thông tin cao nhất làm điểm phân tách đầu tiên trong cây quyết định.
- Trong trường hợp này, **Outlook** tạo ra mức thu được thông tin cao nhất. Từ đó, quá trình này được lặp lại cho mỗi cây con.

4.4. Ưu điểm

- Đơn giản để hiểu và giải thích vì cây có thể được hình dung.
- Yêu cầu chuẩn bị ít dữ liệu. Các kỹ thuật khác thường yêu cầu chuẩn hóa dữ liệu, cần tạo các biến giả và xóa các giá trị trống. Một số cây và thuật toán kết hợp hỗ trợ các giá trị bị thiếu (missing values).
- Chi phí sử dụng cây (tức là dự đoán dữ liệu) là logarit theo số lượng điểm dữ liệu được sử dụng để huấn luyện cây.
- Có khả năng xử lý cả dữ liệu số và dữ liệu phân loại.
- Có khả năng xử lý các vấn đề đa đầu ra (multi-output problems).
- Sử dụng mô hình hộp trắng (white box model). Nếu một tình huống nhất định có thể quan sát được trong một mô hình thì việc giải thích cho điều kiện đó có thể dễ dàng được giải thích bằng logic boolean. Ngược lại, trong mô hình hộp

đen (ví dụ: trong mạng lưới thần kinh nhân tạo), kết quả có thể khó diễn giải hơn.

- Có thể xác nhận một mô hình bằng các bài kiểm tra thống kê. Điều đó làm cho nó có thể giải thích được độ tin cậy của mô hình.
- Hoạt động tốt ngay cả khi các giả định của nó bị vi phạm phần nào bởi mô hình thực mà dữ liệu được tạo ra từ đó.

4.5. Nhược điểm

- Người học có thể tạo ra những cây quá phức tạp mà không khái quát hóa tốt dữ liệu. Điều này được gọi là **overfitting**.
- Cây quyết định có thể không ổn định vì những thay đổi nhỏ trong dữ liệu có thể dẫn đến việc tạo ra một cây hoàn toàn khác. Vấn đề này được giảm thiểu bằng cách sử dụng cây quyết định trong một tập hợp.
- Dự đoán của cây quyết định không trơn tru cũng không liên tục mà là các xấp xỉ không đổi từng phần.
- Bài toán học cây quyết định tối ưu được gọi là “NP-complete” dưới một số khía cạnh của tính tối ưu và đơn giản. Do đó, các thuật toán học cây quyết định thực tế dựa trên heuristic như **thuật toán greedy** trong đó các quyết định tối ưu cục bộ được đưa ra tại mỗi node. Các thuật toán như vậy không thể đảm bảo trả về cây quyết định tối ưu toàn cục.
- Có những khái niệm khó học vì cây quyết định không thể hiện chúng một cách dễ dàng, chẳng hạn như các vấn đề về XOR, tính chẵn lẻ hoặc bộ ghép kênh.
- Người dùng cây quyết định tạo ra “cây thiên vị” (biased trees) nếu một số lớp chiếm ưu thế. Do đó, nên cân bằng tập dữ liệu trước khi khớp với cây quyết định.

Tham khảo

Linear Regression

- <https://www.statisticssolutions.com/free-resources/directory-of-statistical-analyses/how-to-conduct-linear-regression/#:~:text=It%20consists%20of%203%20stages,directionality%20and%20correlation%20of%20data>
- <https://www.analyticsvidhya.com/blog/2021/06/linear-regression-in-machine-learning/#h-what-is-a-regression>
- <https://www.knowledgehut.com/blog/data-science/linear-regression-for-machine-learning#use%20cases-of-linear-regression%20>

Decision Tree

- <https://scikit-learn.org/stable/modules/tree.html#decision-trees>
- <https://www.ibm.com/topics/decision-trees#:~:text=A%20decision%20tree%20is%20a,internal%20nodes%20and%20leaf%20nodes>
- <https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm>

Naive Bayes classifiers

- <https://www.ibm.com/topics/naive-bayes#:~:text=The%20Na%C3%AFve%20Bayes%20classifier%20is,a%20given%20class%20or%20category>
- https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/#Applications_of_Naive_Bayes_Algorithms

kNN

- <https://www.analyticsvidhya.com/blog/2018/08/k-nearest-neighbor-introduction-regression-python/>
- <https://machinelearningcoban.com/2017/01/08/knn/>