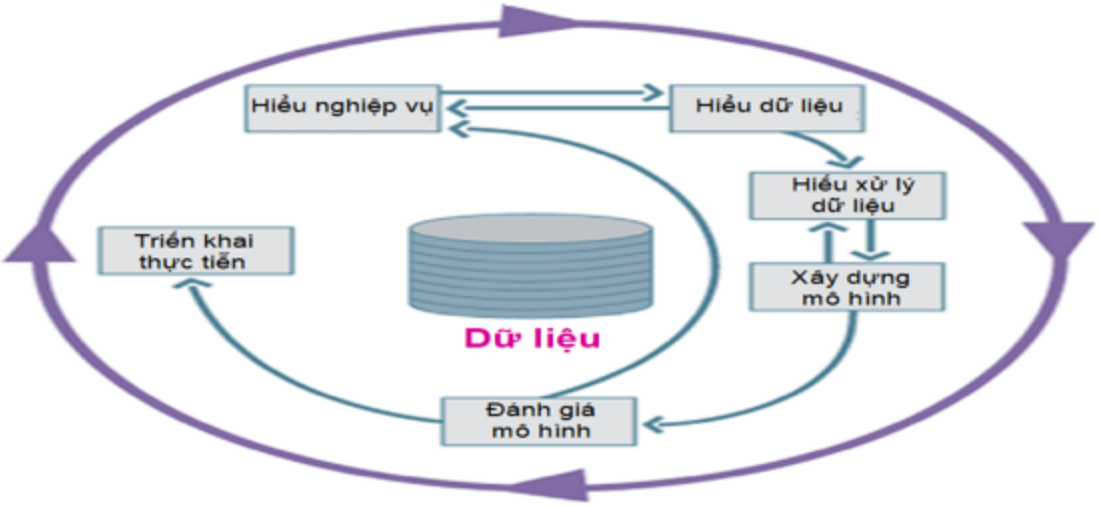
Câu hỏi

Câu 1: Trình bày quy trình mô hình chuẩn công nghiệp khai phá dữ liệu CRISP-DM. Tại sao bước hiểu kinh doanh và hiểu dữ liệu là quan trọng. Nêu điểm giống và khác nhau

(càng nhiều càng tốt) giữa bài toán khai phá dữ liệu và bài toán CSDL.

Quy trình mô hình chuẩn công nghiệp khai phá dữ liệu CRISP-DM:



Thứ tự thực hiện:

- Hiểu kinh doanh, hiểu nghiệp vụ, bài toàn cần phải thực hiện.
- Hiểu dữ liệu.
- Hiểu xử lý dữ liệu.
- Xây dựng mô hình.
- Đánh giá mô hình.
- Đối chiếu lại với nghiệp vụ đã hiểu ở bước 1.
- Triển khau thực tiễn.
- Thực hiện lặp đi lặp lại các bước.

Cụ thể như sau:

- Hiểu kinh doanh: Tập trung vào hiểu biết mục tiêu / yêu cầu từ góc độ kinh doanh.
    - Chuyển đổi các tri thức này thành một định nghĩa bài toàn khai phá dữ liệu.
    - Một kế hoạch sơ bộ được thiết kế để đạt được các mục tiêu.

- Hiểu dữ liệu:
  - Với một tập dữ liệu ban đầu, cần tiến hành hoạt động làm quen dữ liệu, xác định các vấn đề chất lượng của dữ liệu.
  - Khám phá hiểu biết ban đầu tới tập dữ liệu / phát hiện các tập con dữ liệu thú vụ nhằm hình thành giả thuyết cho thông tin ẩn.
  - Tri thức kinh doanh từ giai đoạn hiểu kinh doanh định hướng việc hiểu dữ liệu → phân tích dữ liệu để hiểu dữ liệu có thể phản hồi, phối hợp với nội dung hiểu kinh doanh → làm rõ bài toán khai phá dữ liệu, mục tiêu và kết hoạch thực hiện.
- Chuẩn bị dữ liệu:
  - Gồm mọi hoạt động nhằm xây dựng các tập dữ liệu cuối làm vào cho công cụ mô hình hoá.
  - Gồm các hoạt động như lập bảng, ghi lại và lựa chọn các thuộc tính, chuyển đổi và làm sạch dữ liệu cho công cụ mô hình hoá.
  - Thực hiện nhiều lần và không theo một thứ tự nhất định.
- Xây dựng mô hình:
  - Các kỹ thuật mô hình khác nhau được lựa chọn và áp dụng vào bài toán để mô hình hoá tốt được bài toán.
  - Xác định các tham số mô hình nhằm đạt tới giá trị tối ưu.
  - Một số kỹ thuật được sử dụng để tăng độ hiệu quả.
  - Lặp lại một số lần mô hình hoá và chuẩn bị dữ liệu nhằm đạt được mô hình có kết quả tối ưu.
- Đánh giá:
  - tìm ra một số mô hình kết quả với mục tiêu chất lượng cao theo góc độ phân tích dữ liệu.
  - Đánh giá mô hình kêt quả kỹ lưỡng và xem xét các bước đã được thực hiện để xây dựng mô hình → niềm tin rằng mô hình kết quả đạt được các mục tiêu kinh doanh theo đúng cách thức.

Quy trình mô hình chuẩn công nghiệp - KPDL

- Đầu tiên, ta cần phải hiểu kinh doanh: hiểu bài toán cần xử lý theo góc độ kinh doanh, phát biểu nó thành một bài toán KPDL.
- Sau đó, cần phải hiểu dữ liệu: Kiểm tra về chất lượng và các mẫu trong dữ liệu, kết hợp với tri thức từ hiểu kinh doanh để hiểu rõ bài toán.
- Chuẩn bị dữ liệu: Chuẩn bị dữ liệu để sử dụng trong xây dựng mô hình, ví dụ như làm sạch, chọn thuộc tính. Cần thực hiện lặp đi lặp lại để có được kết quả tốt nhất.
- Xây dựng mô hình: Sử dụng các mô hình phổ biến cùng dữ liệu đã chuẩn bị, tham số được tính chỉnh để có được kết quả tối ưu. Hai pha chuẩn bị dữ liệu và xây dựng mô hình cần thực hiện lặp đi lặp lại để có hiệu suất cao.
- Đánh giá mô hình: Sử dụng các mô hình và đánh giá chất lượng của mô hình, xem xét các bước xây dựng mô hình. Sau đó, quay lại bước hiểu kinh doanh xem đã hiểu rõ bài toán chưa.
- Cuối cùng là triển khai thực tiễn.

Lí do hiểu kinh doanh và hiểu dữ liệu quan trọng

Hai pha hiểu kinh doanh và hiểu dữ liệu là cực kì quan trọng. Lí do là bởi vì hai pha này nằm ở phần đầu của toàn bộ quy trình, có vai trò tìm hiểu và xác định các yêu cầu của toàn bộ bài toán. Việc này sẽ quyết định mạnh mẽ đến hướng phát triển của toàn bộ quy trình, đặc biệt là hai bước chuẩn bị dữ liệu và xây dựng mô hình. Nếu như không tìm hiểu một cách chính xác, chuẩn bị dữ liệu sẽ không thể xử lý dữ liệu theo đúng hướng, và mô hình xây dựng được cũng không phục vụ được yêu cầu của bài toán. Và khi đó, kết quả thu được là không đáp ứng yêu cầu nghiệp vụ đặt ra.

So sánh bài toán KPDL và bài toán CSDL

Giống nhau:

- Cả hai đều là những bài toán liên quan tới việc xử lý dữ liệu và truy xuất kết quả cho người dùng.
- Đa dạng dữ liệu đầu vào và đầu ra.
- Dung lượng dữ liệu cực lớn.

Khác nhau:

- Yêu cầu của bài toán CSDL là tìm ra các mẫu cần thiết ở trong Database và cung cấp cho người dùng. Ví dụ như là tìm ra số tiền của người A vào ngày 15 tháng 5 năm 2023, hay lượng cổ phiếu với mệnh giá tăng trong ngày. Trong khi đó, bài toán KPDL hướng tới việc tìm hiểu được tri thức từ database, tức là từ dữ liệu và mẫu có được, các mô hình sẽ phân tích và tìm ra được tri thức có trong các mẫu đó, và đưa ra những đặc điểm mang tính quy luật của dữ liệu. Ví dụ như cổ phiếu tăng hôm nay có những đặc trưng gì, Hy vọng gì trong tuấn tiếp theo, …
- Bài toán CSDL yêu cầu độ chính xác cao, do yêu cầu về các mẫu có sẵn trong database. Trong khi đó, bài toán KPDL chỉ cần những giá trị mang tính ước lượng, tính chính xác không cần quá cao, vì yêu cầu là tri thức từ database.
- Bài toán CSDL có tính đầy đủ hơn về miền tri thức, trong khi đó bài toán KPDL thì giả thiết về tính đầy đủ là không còn, cần bổ sung tri thức cho hệ thống → cải tiến, nâng cấp miền tri thức.

Câu 2:

Giải thiết bài toán

Ta có tập các mục (mặt hàng): I = {i1, i2, i3, …, ik}.

Định nghĩa một giao dịch T sẽ bao gồm nhiều mục, là tập con của I.

Cơ sở dữ liệu giao dịch: D = {T : T là tập con của I}. Mỗi giao dịch chưa định danh Tid.

A là một tập các mục bất kì. A là tập con của I. Nếu T chứa A, tức A là con của T.

Luật kết hợp

Luật kết hợp là mối quan hệ giữa các tập mục trong cơ sở dữ liệu. Luật kết hợp là một phương tiện hữu ích để khám phá ra các mối liên kết trong dữ liệu.

Luật kết hợp có dạng X→Y, trong đó X và Y là hai tập con của I, không giao nhau. X và Y là hai tập mục, tương tự với A ở giả thiết. Luật kết hợp có ý nghĩa là khi có tập mục X thì có tập mục Y → thể hiện mối quan hệ giữa hai tập mục.

Độ đo liên quan

Độ hỗ trợ

Độ hỗ trợ của một tập mục A là xác suất xuất hiện tập A các giao dịch thuộc CSDL.

Độ hỗ trợ của luật kết hợp X→Y là xác suất xuất hiện cả X và Y trong giao dịch thuộc CSDL.

$$s(X \to Y) = \frac{|\{T \in D : X \cup Y \subseteq T\}|}{|D|}$$

$$1 \geq s\,(X \to Y) \geq 0$$

Độ tin cậy

Độ tin cậy của luật kết hợp X→Y là tỷ lệ giữa số giao dịch chứa cả (X và Y) so với lượng giao dịch chứa X.

$$c(X \to Y) = \frac{S(X \to Y)}{S(X)}$$

$$1 \geq c\,(X \to Y) \geq 0$$

Tập phổ biến

A là một tập phổ biến khi độ hỗ trợ s(A) lớn hơn một độ hỗ trợ tối thiểu s0 > 0 nào đó.

⇒ s(A) ≥ s0.

Hai bước để tìm luật kết hợp trong một CSDL giao dịch

- Bước 1: Tìm mọi tập mục phổ biến trong CSDL D với ngưỡng độ hỗ trợ tối thiểu s0 cho trước.
- Bước 2: Sinh ra tất cả các luật mạnh từ các tập phổ biến khai phá được từ pha trước với ngưỡng tin cậy c0 cho trước.

Câu 3:

Phát biểu bài toán tìm tập phổ biến và luật manh

Bài toán tìm tập phổ biến

Cho trước một CSDL giao dịch D, độ hỗ trợ tối thiểu là s0 > 0.

Tìm tất cả các tập phổ biến từ D: {A là con của I: s(A) ≥ s0}.

Bài toán tìm luật mạnh

Cho trước một CSDL giao dịch D, độ hỗ trợ tối thiểu là s0 > 0 và độ tin cậy tối thiểu c0 > 0.

Tìm tất cả các luật mạnh từ D: {A→B: s(A→B) ≥ s0, c(A→B) ≥ c0}.

Thuật toán Apriori tìm tập phổ biến

Input

- Cơ sở dữ liệu giao dịch D = {t, t là giao dịch}.
- Độ hỗ trợ tối thiểu minsup > 0.

Output

- Tập hợp tất cả các tập phổ biến.

Nội dung chính

Cơ sở của thuật toán:

- Mọi tập con của một tập phổ biến cũng là tập phổ biến.
- Nguyên lý cắt tỉa Apriori: Với các tập mục không phổ biến thì không cần phải sinh ra ứng viên hoặc kiểm tra mọi tập bao tập mục đó. Tức là ta sẽ chỉ xét các tập mục phổ biến và sinh ra các ứng viên tập phổ biến từ nó thôi.

Các bước chính

- Sử dụng phương pháp quy hoạch động dựa theo độ dài của tập mục phổ biến.
- Tìm tất cả các tập phổ biến có độ dài là 1.
- Xét độ dài k.

- Tìm ra tất cả các tập phổ biến có độ dài k, kí hiệu là F[k].
- Để tìm tập F[k+1]:
  - Sinh ứng viên: Tạo tập C[k+1] là mọi tập ứng viên có độ dài k+1 từ tập F[k].
    - Bước ghép: Sinh ra tập c[k+1] từ hai tập mục P[k] và Q[k] thuộc F[k].
      - $P_k = \{i_1, i_2, \dots, i_{k-1}, i_k\}$,
      - $Q_k = \{i_1, i_2, \dots, i_{k-1}, i_{k'}\}$,
        $c_{k+1} = \{i_1, i_2, \dots, i_{k-1}, i_k, i_{k'}\}; i_1 < i_{2<}, \dots < i_{k-1} < i_k < i_{k'}$
    - Bước tỉa: Xét lại mọi X thuộc c[k+1], |X| = k, nếu mọi X thuộc F[k] thì lấy c[k+1] làm ứng viên (do mọi tập con của tập phổ biến cx phải là tập phổ biến), ngược lại thì bỏ qua.
  - Duyệt CSDL D để lọc ra các tập phổ biến thuộc F[k+1] từ C[k+1].

Giải thích sơ qua về tính đúng đắn của thuật toán

- Thuật toán sử dụng việc quy hoạch động theo độ dài của tập phổ biến, với F[k] là các tập phổ biến có độ dài k.
- Ý tưởng của thuật toán là khi sinh ứng viên C[k+1] từ F[k]. Đúng đắn vì khi f[k+1] là một tập phổ biến độ dài k + 1, thì mọi tập con của nó, có độ dài từ 1 đến k + 1 đều phải là tập phổ biến. Giả sử f[k+1] là tập phổ biến, và fk là con của f[k+1] có độ dài k. Vì f[k+1] là tập phổ biến, nên các tập trong fk chắc chẵn cũng phải có lần xuất hiện trong CSDL D là lớn hơn so với mức tối thiểu. Do đó nó có độ hỗ trợ lớn hơn độ hỗ trợ tối thiểu. ⇒ fk là tập phổ biến. Và bởi vì fk là tập phổ biến nên tương tự các con của nó cũng là tập phổ biến.

⇒ Tất cả các tập con của f[k+1] đều là tập phổ biến.

- Việc sinh tập ứng viên từ tập phổ biến chưa hẳn đã có thể mang lại luôn một tập phổ biến mới, do việc ghép 1 tập vào 1 tập phổ biến chưa đảm bảo hợp của chúng là tập phổ biến. Vì vậy, cần có bước tỉa để loại bỏ các ứng viên mà tập con không phải là tập phổ biến. ⇒ Đảm bảo các Tập sau bước tỉa là các tập phổ biến.

⇒ Các F[k] sau khi xử lý sẽ bao gồm các tập phổ biến.

Câu 4:

Bài toán phân lớp

Phát biểu: Cho miền dữ liệu D, tập phân lớp C (Mỗi dữ liệu dj thuộc D thuộc một phân lớp Ci nào đó), tập dữ liệu D_train (gồm một example và một phân lớp tương ứng với example đó). Sử dụng tập dữ liệu D_train để huấn luyện mô hình ánh xạ xấp xỉ tốt nhất từ D → C.

Đầu vào:

- Tập dữ liệu D = {dj}.
- Tập các phân lớp C = {C1, C2, C3, …}. Mỗi dj sẽ thuộc một phân lớp Ci nào đó.
- Tập ví dụ D_exam = {D1, D2, D3, …, Dk} đại diện cho tập dữ liệu D.
- D gồm m dữ liệu dj thuộc không gian n chiều.

Đầu ra:

- Mô hình phân lớp: Ánh xạ từ D sang C.

Để kiểm tra, sử dụng các example không thuộc D_exam.

Phân lớp Bayes

Cơ sở lý thuyết

- Định lý Bayes:
  - Giả sử có hai biến cố X và C.
  - Xác suất có điều kiện P(X/C) = P(X,C) / P(C), P(C/X) = P(X,C) / P(X).

  ⇒ P(C/X) = P(X/C)*P(C) / P(X).
- Áp dụng vào bài toán phân lớp:
  - X là input của bài toán.
  - C là tập các phân lớp.
  - P(Ci): Tần suất xuất hiện của dữ liệu thuộc phân lớp Ci.
  - P(Ci/X) là xác suất khi có input X và rơi vào phân lớp Ci.

  ⇒ Yêu cầu bài toán ↔ Tìm Ci sao cho P(Ci/X) là lớn nhất.

Giả thiết

- Các đặc trưng đưa vào mô hình là độc lập với nhau, sự thay đổi của đặc trưng này không ảnh hưởng đến các đặc trưng còn lại.
- Các đặc trưng đưa vào mô hình có ảnh hưởng ngang nhau tới đầu ra mục tiêu.

Mô hình phân lớp

Cho Input X là một vector gồm các đặc trưng của một đối tượng:

$$X = (x_1, x_2, x_3, ..., x_n)$$

Khi đó, đẳng thứ Bayes với Output phân lớp Y và Input X P(y / X) có dạng:

$$P(y|x_1, ..., x_n) = \frac{P(x_1|y)P(x_2|y)...P(x_n|y)P(y)}{P(x_1)P(x_2)...P(x_n)}$$

Khi đó, kết quả mục tiêu cần tìm của bài toán phân lớp này là một phân lớp y nào đó sao cho P(y/X) đạt giá trị lớn nhất. Hay:

$$y = argmax_y P(y) \prod_{i=1}^{n} P(x_i|y)$$

Giải pháp chính

- Gọi D là tập dữ liệu huấn luyện, mỗi phần tử dữ liệu x được biểu diễn bởi một vector chứa n giá trị thuộc tính {x1, x2, …, xn}
- Có m phân lớp C1, C2, …, Cm. Mỗi phần tử thuộc tập dữ liệu X được bộ phân lớp gán nhãn cho lớp có xác suất xảy ra khi có Input X là lớn nhất. Tức là X sẽ được gán nhãn là thuộc Ci khi P(Ci / X) max.
- Để tính được xác suất trên, ta thấy rằng P(X) bằng nhau với mọi Ci nên có thể bỏ qua phần đó. Còn P(Ci) ta cũng có thể coi như là P(C1) = P(C2) = … = P(Cm) khi

xác suất P(Ci) không xác định. Do đó, thứ ta cần tìm là giá trị Ci sao cho P(X/Ci) là lớn nhất.

- Khi đó, P(X/Ci) = P(x1/Ci) x P(x2/Ci) x … x P(xn/Ci).

Ưu điểm của phân lớp Bayes

- Giả định độc lập của phân lớp Bayes có thể hoạt động tốt cho nhiều bài toán / miền dữ liệu và ứng dụng.
  - Đủ tốt để giải quyết các bài toán như phân lớp văn bản, lọc spam.
- Cho phép kết hợp tri thức tiền nghiệm và dữ liệu quan sát được.
  - Tốt khi có sự chênh lệch số lượng giữa các lớp phân loại.
- Huấn luyện mô hình (ước lượng tham số) dễ và nhanh.

Nhược điểm

- Giả định độc lập không quá tốt khi hầu như các trường hợp thực tế, các thuộc tính thường phụ thuộc vào nhau.
- Vấn đề zero.
- Mô hình không được huấn luyện bằng phương pháp tối ưu mạnh và chặt chẽ.
- Tham số của mô hình là các ước lượng xác suất điều kiện đơn lẻ, không tính đến sự tương tác giữa các ước lượng này.

Câu 5:

Phân lớp cây quyết định

Mô hình phân lớp: Cây quyết định

Cây quyết định là cây được gán nhãn ở cả nút lẫn cung.

- Lá: Giá trị lớp (nhãn), có một cung vào, không có cung ra.
- Nút trong: Tên thuộc tính, có chính xác một nút vào hoặc nhiều nút ra (gắn với điều kiện kiểm tra giá trị thuộc tính ở nút đó).
- Gốc: Không có cung vào và 0 hoặc nhiều cung ra, tương tự với nút trong.

Dữ liệu đầu vào sẽ được phân chia thành các tập con qua từng nút của cây theo điều kiện được quy định ở các cung. Khi các dữ liệu sau một vài bước được phân chia tới nút lá thì gán nhãn tại lá cho dữ liệu đó.

Cơ sở lý thuyết

Giải pháp chính

Xây dựng cây quyết định

- Phương châm “chia để trị”. Tại mỗi nút sẽ chứa một tập dữ liệu nhất định. Dựa vào giá trị của trường thuộc tính trên nút đó của các dữ liệu mà chúng sẽ được phân chia xuống các nút ở phía dưới.
- Đệ quy theo node của cây.
- Xét nút thứ t trên cây. Giả sử node đó đang có các example Dt trên nút đó. Tập nhãn hiện tại là y1, y2, y3, …, yk (k phân lớp). Và ta cần phải xác định được nhãn và các cung ra của nút này.
- Nếu mọi example trong Dt đều thuộc có cùng một nhãn yi thì kết luận t là lá, và gán nhãn yi.
- Nếu các example trong Dt thuộc nhiều phân lớp khác nhau thì:
    - Chọn 1 thuộc tính A bất kì nào đó để phân hoạch Dt theo thuộc tính đó. Gán nhãn A cho nút.
    - Phân hoạch Dt theo tập giá trị của A thành các tập con nhỏ hơn.
    - Với mỗi tập con theo phân hoạch ở trên sẽ tương ứng với một nút con u của t. Cung nối u và t có giá trị là giá trị của trường A của tập con đó.
    - Cứ thế lặp đi lặp lại quá trình với từng nút con u cho tới khi các example thuộc cùng phân lớp hoặc đến độ sâu nhất định.
- Tuy nhiên, việc chọn ra được các thuộc tính A tại các nút lại là việc khó khăn và cần phải tối ưu hoá sao cho nó có thể mang lại kết quả tốt nhất ⇒ chọn thuộc tính là chiến lược tối ưu hoá.
- Để có thể chọn ra được thuộc tính tốt nhất, có thể sử dụng độ đo Gini hoặc Information Gain. Cần phải tìm ra được thuộc tính A sao cho Gini(t) là nhỏ nhất và Information Gain là lớn nhất.

Sử dụng cây quyết định:

- Ta có dữ liệu X với các thuộc tính X = {x1, x2, …, xn}.
- Bắt đầu từ nút gốc, xét giá trị thuộc tính A của X, đưa X tới nút con tương ứng với giá trị đó.
- Tại các nút trong, xét giá trị thuộc tính At tại nút đó, và đưa X tới nút con của t tương ứng với giá trị đó.
- Tiếp tục cho tới khi nào gặp được nút lá và gán nhãn cho X.

Ưu điểm

- Dễ hiểu hơn, trực quan hoá quá trình phân lớp.
- Việc chuẩn hoá dữ liệu không quá cần thiết.
- Có thể xử lý cả dữ liệu có giá trị số hoặc tên thể loại.
- Là một mô hình hộp trắng.
- Xử lý được lượng dữ liệu lớn trong thời gian ngắn.

Nhược điểm

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

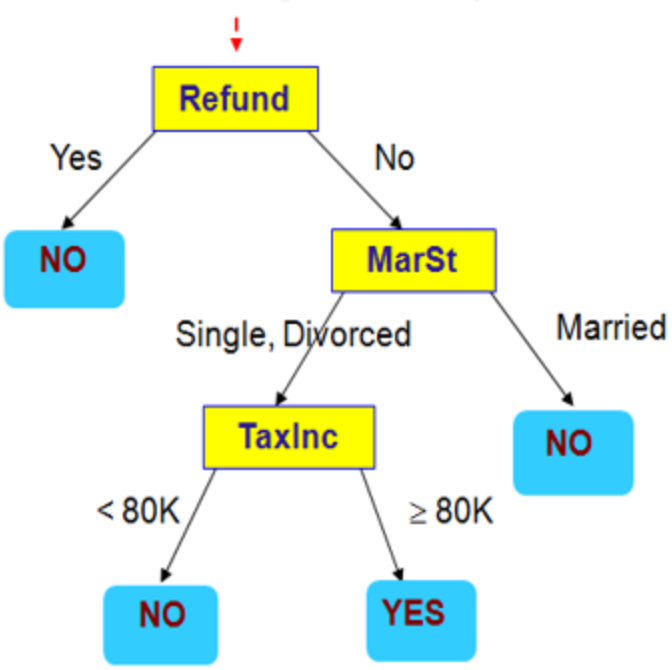
Ví dụ

Giả sử bài toán phân loại xem có khả năng khách hàng gian lận ngân hàng hay không?

Đầu vào là các đặc trưng của khách hàng: Refund, Marital Status, Taxable Income.

Đầu ra là phân loại xem khách hàng này có khả năng gian lận hay không?

Ta có cây quyết định như sau:



Giả sử có một khách hàng có thông tin:

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Thì ta sẽ thực hiện gán nhãn như sau:

- Đi như gốc, thuộc tính Refund = No ⇒ Đi xuống con bên phải.
- Thuộc tính Marital Status = Married ⇒ Đi sang con bên phải.
- Gặp nút lá ⇒ Gán nhãn No cho khách hàng ⇒ Khách hàng không gian lận.

Câu 6:

K láng giềng gần nhất (KNN)

Cơ sở lý thuyết

- Giả định rằng các điểm dữ liệu gần nhau trong không gian đặc trưng sẽ có cùng nhãn.
- Sự tương đồng giữa các điểm dữ liệu được đo bằng các độ đo khoảng cách.

Giải pháp chính

- Lưu trữ giữ liệu huấn luyện trong tập D, mỗi dữ liệu X bao gồm nhiều thuộc tính {x1, x2, …, xn}.
- Sử dụng một độ đo nhất định (Euclid, cos, sin, …) để đo khoảng cách giữa điểm dữ liệu cần phân lớp và tất cả các điểm dữ liệu huấn luyện.
- Chọn ra k điểm dữ liệu gần nhất (K láng giềng gần nhất) dựa trên khoảng cách đã tính ở trên.
- Sử dụng đa số phiếu bầu để quyết định nhãn của điểm dữ liệu cần phân lớp. Tức là nhãn phổ biến nhất trong K láng giềng gần nhất sẽ được gán cho điểm dữ liệu cần phân lớp.

Ưu điểm

- Đơn giản dễ hiểu.
- Xử lý đa dạng dữ liệu: Áp dụng cho cả dữ liệu rời rạc và liên tục.
- Tính diễn giải cao: Kết quả của KNN có thể được dễ dàng giải thích và diễn giải bằng cách xem xét K làng giềng gần nhất.
- Hiệu suất tốt với dữ liệu nhỏ.

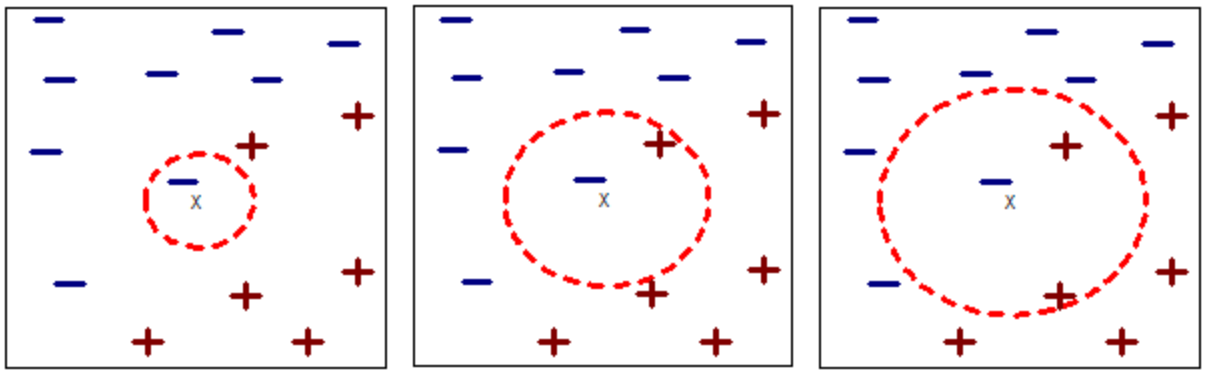
Nhược điểm

- Tính toán phức tạp: Do phải tính toán khoảng cách từ điểm dữ liệu cần phân lớp và tất cả các điểm trong tập huấn luyện ⇒ Tốn kém về mặt tính toán và đối với tập dữ liệu lớn.
- Yêu cầu lưu trữ toàn bộ dữ liệu huấn luyện: Do mỗi lần phân lớp đều cần tính khoảng cách giữa các điểm rồi mới chọn ra k láng giềng gần nhất, nên cần lưu trữ tất cả các điểm.
- Nhạy cảmvới nhiễu và dữ liệu không cân bằng.

- Khó khăn trong lựa chọn tham số K: Quá nhỏ gây ra overfitting, quá lớn gây ra underfitting.

Ví dụ

Giả sử có hai phân lớp là + và - trong không gian thuộc tính. Ta biểu diễn dữ liệu như sau:



3 hình trên tương tự với 3 trường hợp giá trị của k là 1, 2 và 3.

Ta có:

- Trường hợp 1, k = 1, khi đó ta gán X với nhãn của láng giềng gần nhất của nó.
- Trường hợp 2,. k = 2, gán X với nhãn phổ biến nhất trong 2 láng giềng.
- Trường hợp 3, k = 3, gán X với nhãn phổ biến trong 3 láng giềng.

Câu 7:

Câu 8:

Bộ độ đo hồi tưởng - chính xác (Recall - Precision)

Độ hồi tưởng (Recall), kí hiệu là p, hoặc TPR (True Positive Rate):

R = TPR = TP / (TP + FN). Thể hiện trong số các dữ liệu nhãn dương thì hệ thống dự đoán đúng được bao nhiêu ⇒ Khả năng dự đoán đúng các nhãn dương.

Độ chính xác (Precision), hay pi thể hiện phần trăm dự đoán đúng của hệ thống: Trong số dự đoán dương thì có bao nhiêu phần trăm là đúng.

P = TP / (TP + FP).

F1-score = 2 /((1 / P) + (1/R)).

Bộ độ đo chính xác - hệ số lỗi (Accuracy và Error rate)

Accuracy = số dự báo chính xác / tổng số dự báo = (TP + TN) / (TP + TN + FP + FN).

Error = số dự báo sai / tổng số dự báo = (FN + FP) / (TP + TN + FP + FN).

So sánh hai độ đo

- Precision - Recall thường được sử dụng trong các bài toán không bằng lớp, số lượng mẫu không đồng đều. Cung cấp thông tin về khả năng phân loại chính xác các điểm dữ liệu và khả năng phát hiện tất cả các điểm dữ liệu thuộc lớp cần quan tâm. Hay nói cách khác, Precision-Recall hướng tới một phân lớp cụ thể, đo độ chính xác khi phân lớp đó.
- Accuracy-Error Rate thường được sử dụng trong các bài toán có sự cân bằng lớp, trong đó các lớp có số lượng mẫu tương đối đồng đều. Accuracy đo lường khả năng phân loại chính xác tổng thể, trong khi Error Rate đo lường tỷ lệ phân loại sai tổng thể. Hay nói cách khác, Accuracy - Error Rate sẽ hướng tới việc phân loại tổng thể, bao quát tất cả các phân lớp.

Câu 9:

Bài toán phân cụm

- Cho tập dữ liệu D = {dj, j = 1 → n}.
- Phân các dữ liệu thuộc D thành các cụm khác nhau.
  - Các dữ liệu trong cùng 1 cụm sẽ gần nhau (tương tự nhau).
  - Các dữ liệu trong các cụm khác nhau sẽ xa nhau (không tương tự nhau).
- Sử dụng một số độ đo để đo độ tương tự nhau của các điểm.

Thuật toán K-mean gán cứng

Input

- Tập dữ liệu D = {d}
- Độ đo tương tự SIM
- k > 0 là số lượng cụm.

Output

- Tập k cụm {(ci, Ci)} (i = 1, k) với ci là tâm cụm i và Ci là tập dữ liệu cụm i.

Thuật toán chính

- Khởi động:
  - Chọn ngẫu nhiên các tâm của cụm {ci thuộc D, i = 1, k}.
- Lặp lại các bước sau:
  - Gán Ci = rỗng, mọi i.
  - Với mọi dữ liệu d thuộc D
    - Tính sim(d, ci), với i = 1, k (Tìm ra xem d gần với tâm cụm nào nhất).
    - Thêm d vào phân cụm Ci, với i là phân cụm có tâm gần với d nhất.
  - Tính lại tâm cụm ci = 1/|Ci| * Tổng (d), d thuộc Ci. (Điều chỉnh các tâm cụm tới điểm chính giữa của cụm bằng cách tính trung bình các dữ liệu trong cụm đó).
- Nếu gặp điều kiện dừng thì kết thúc, còn không quay lại 2 q.

Lưu ý

- Kết quả sau khi tính lại tâm cụm có thể không trùng với điểm dữ liệu nào mà chỉ đơn giản là vị trí chính giữa của cụm.
- Dùng độ đo khoảng cách thay vì độ đo tương tựu.
- Điều kiện dừng:
  - Sau bước 2 cụm không thay đổi.
  - Khống chế số lần lặp.

- Độ đo Cost Function tới đủ nhỏ (Trung bình khoảng cách từ các điểm dữ liệu tới tâm của phân cụm của điểm dữ liệu đó).

Ưu điểm

- Đơn giản, dễ sử dụng
- Hiệu quả về thời gian: Tuyến tính, O(tkn), t là số lần lặp, k là số cụm, n là số phần từ.
- Một thuật toán phân cụm phổ biến nhất.
- Thường cho tối ưu cụ bộ: tối ưu toàn cục rất khó tìm.

Nhược điểm

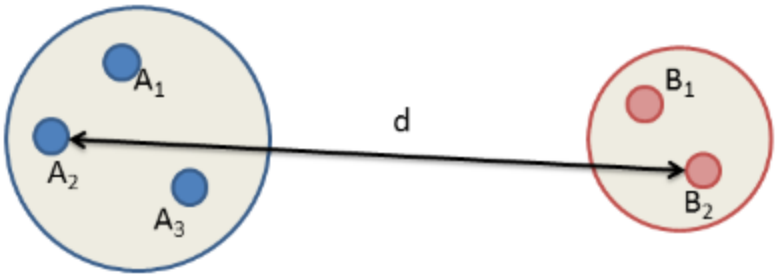
- Phải “tính trung bình được”: Dữ liệu phân lớp thì dựa theo tần số.
- Cần cho trước k: số cụm.
- Nhạy cảm với ngoại lệ (có thể có một điểm lạ nào đó cách xa với cụm thì khó có thể phân cụm chính xác được).
- Nhạy cảm với mẫu ban đầu.
- Không thích hợp với các tập dữ liệu không siêu ellip hoặc siêu cầu.

Câu 10:

Độ đo tương tự cụm

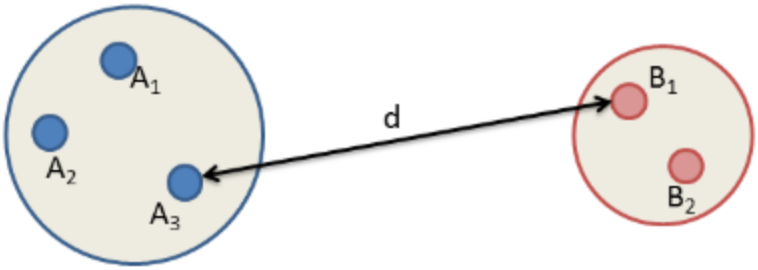
Độ tương tự cực tiểu (complete-link)

Độ tương tự cực tiểu được tính bằng khoảng cách giữa cặp phần tử xa nhất của trong hai phân cụm.



Độ tương tự cực đại (single-link)

Độ tương tự cực đại là khoảng cách giữa cặp phần tử gần nhất trong hai phân cụm.



Phân cụm phân cấp (HAC)

Input

- Tập dữ liệu D = {d}, độ đo tương tự sim.

Output

- Tập các cụm phân cấp của D.

Thuật toán

- Khởi tạo G = {d | d thuộc D} ⇒ Khởi tạo G là tập các phân cụm có một phần tử dữ liệu.
- Lưu lại G.
- Tìm i, j sao cho (i, j) = argmax(u, v) sim(Cu, Cv) ⇒ Tìm ra hai cụm Ci và Cj sao cho chúng có tương tự với nhau nhất.

- Loại bỏ Ci, Cj, và thêm vào G Ci hợp Cj.
- Nếu |G| = 1 thì dừng lại, còn không quay lại bước 2.

Lưu ý

- Nếu muốn phân cụm cho tới khi còn k cụm thì đặt điều kiện |G| = k.
- Có thể tính trước khoảng cách giữa các cụm. Khi có cụm mới thêm vào thì tính toán luôn khoảng cách từ cụm mới tới các cụm cũ.

Ưu điểm

- Có thể phân cụm với mọi giá trị của K.
- Được sử dụng thường để xác định, dự đoán số cụm sao cho tốt nhất trước khi sử dụng Kmean.

Nhược điểm

- Không thuận tiện cho tập dữ liệu khổng lồ.

Câu 11:

Các phương pháp đánh giá thuật toán phân cụm

Thực trạng: Đánh giá chất lượng của mô hình phân cụm là khó khăn khi chưa biết được số cụm thực sự.

Một số phương pháp

- Người dùng kiểm tra:
  - Nghiên cứu trọng tâm và miền phủ.
  - Luật từ cây quyết định.
  - Đọc các dữ liệu trong cụm.
- Đánh giá theo các độ đo tương tự / khoảng cách
  - Độ phân biệt giữa các cụm.
    - Cực đại hoá tổng tương tự nội tại của các cụm.

    - Cực tiểu hoá tổng độ tương tự các cặp cụm khác nhau.
    - Lấy độ tương tự cực tiểu (complete link) và cực đại(single link).

$$J_e = \frac{1}{2}\sum_{i=1}^{k}\frac{1}{|S_i|}\sum_{d_j,d_l\in S_i}\|d_j - d_l\|^2$$

$$J_s = \frac{1}{2}\sum_{i=1}^{k}\frac{1}{|S_i|}\sum_{d_j,d_l\in S_i} sim(d_j,d_l) = \frac{1}{2}\sum_{i=1}^{k}|S_i|sim(S_i)$$

  - Phân ly theo trọng tâm.

$$J_e = \sum_{i=1}^{k}\sum_{d\in S_i}\|d - c_i\|^2$$

- Dùng thuật toán phân lớp.
  - Coi mỗi cụm là một lớp.
  - Học bộ phân lớp đa lớp.
  - Xây dựng ma trận nhầm lẫn khi phân lớp.
  - Tính các độ đo, để đánh giá theo độ đo đó.

Câu 12:

Hệ thống tư vấn

- Hệ thống tư vấn là các công cụ phần mềm và kỹ thuật cung cấp các tư vấn về các mục có khả năng cao là hữu ích nhất đối với một người dùng đích nào đó.
- Trong hệ tư vấn gồm có:
  - Danh sách về người dùng U, kích thước m.
  - Danh sách các item I (sản phẩm, bài viết, trang web, bản nhạc, …), kích thước n.

- Ma trận hữu ích P:
  - Kích thước m x n ghi lại mức độ hữu ích của item đối với người dùng.
  - p(i, j) ghi lại mức độ hữu ích của item j đối với user i.
  - Giá trị p(i, j) có thể là đã biết hoặc chưa biết. Đã biết là các giả thiết của bài toán, là những gì mà người dùng đã đánh giá.
  - Giá trị chưa biết là những gì mà hệ tư vấn cần phải dự đoán.

Tính chất của hệ tư vấn

- Tính có liên quan:
  - Các mục tư vấn cần liên quan tới người dùng.
- Tính mới lạ:
  - Tư vấn các mục người dùng chưa hoặc khó quan sát.
  - Tránh tư vấn lặp các mục có tính phổ biến.
- Tính may mắn bất ngờ:
  - Tạo ngạc nhiên cho người dùng.
  - Không chỉ là chưa quan sát được.
- Tính đa dạng gia tăng:
  - Các mục tư vấn cần đa dạng, tránh thuộc cùng một thể loại.
  - Lựa chọn tư vấn mục cùng thể loại theo tư vấn khác nhau.
- Tính giải trình:
  - Nên có giải trình mục được tư vấn.

So sánh lọc cộng tác và phân lớp

Phân lớp:

- Chia ra dữ liệu học và dữ liệu đánh giá.
- Chia ra các biến độc lập và biến phụ thuộc

Lọc cộng tác

- Không chia ranh giới giữa dữ liệu học và dữ liệu đánh giá, được biểu diễn trung trên ma trận hữu ích.
- Khong chia ranh giới biến phụ thuộc với biến độc lập.

Câu 13:

Hệ thống kỹ thuật lọc trong hệ thống tư vấn

Kỹ thuật hệ thống tư vấn:

- Kỹ thuật dựa trên nội dung.
- Kỹ thuật dựa trên cộng tác
  - Kỹ thuật dựa trên mô hình:
    - Phân cụm.
    - Luật kết hợp.
    - Mạng Bayes, …
  - Kỹ thuật dựa trên ghi nhớ:
    - Hướng người dùng.
    - Hướng mục.
- Kỹ thuật dựa trên tri thức.
- Kỹ thuật dựa trên nhân khẩu học.
- Kỹ thuật kết hợp.

Tư vấn xã hội

- Định nghĩ hẹp: Là hệ tư vấn sử dụng các mối quan hệ xã hội như đầu vào để bổ sung cho hệ tư vấn hiện tại, sử dụng thêm các tín hiệu xã hội. Các mối quan hệ ở đây có thể là thông tin trao đổi từ những người bạn, từ, quan hệ thành viên, quan hệ tin cậy, …
- Định nghĩa rộng: Là hệ tư vấn sử dụng mọi dữ liệu từ phương tiện xã hội, nhắm đến là mọi đối tượng, lĩnh vực trong xã hội ngày nay.

Tư vấn nhóm

- Là một hệ tư vấn hướng tới một nhóm đối tượng có các mối quan tâm chung cần tư vấn. Các đối tượng trong nhóm có thể tương tác thường xuyên và ảnh hưởng tới quyết định của đối tượng khác.
- Hệ tư vấn nhóm sẽ giải quyết vấn đề đưa ra một tư vấn tốt nhất cho một nhóm, sao cho có thể thoả mãn mọi người với mỗi sở thích và nhu cầu khác nhau.
- Tư vấn nhóm đưa ra gợi ý phản ánh các mối quan tâm và sở thích của tất cả các thành viên trong nhóm ở mức cao nhất, có thể bằng điểm số tính toán cho những gợi ý dựa trên sở thích và đặc điểm của mọi người trong nhóm.

Câu 14:

Lọc cộng tác hướng người dùng

Sơ đồ khối

Giải thích

- Cho bài toán ta đang cần dự đoán độ hữu ích của các Item đối với User U.
- Đầu tiên, ta sẽ tính toán độ tương tự của User U với các User khác bằng công thức độ tương tự:

$$PC(u,v) = \frac{\sum_{i \in S_{uv}} (p_{ui} - \bar{p}_u) \times (p_{vi} - \bar{p}_v)}{\sqrt{\sum_{i \in S_{uv}} (p_{ui} - \bar{p}_u)^2} \times \sqrt{\sum_{i \in S_{uv}} (p_{vi} - \bar{p}_v)^2}}$$

- Trong đó, Suv = Su giao Sv, là tập các Item mà cả U và V đều đánh giá. Qua độ tương tự xác định các láng giềng của U là tập N(U).
- P_(V) là giá trị trung bình các đánh giá của người dùng U.
- Để tính toán độ hữu ích của một Item i với người dùng U, ta tính như sau:

$$p_{u,i} = \bar{p}_u + \frac{\sum_{v \in N(u)} sim\ (p_u, p_v)(p_{v,i} - \bar{p}_v)}{\sum_{v \in N(u)} sim\ (p_u, p_v)}$$

- Dễ thấy vì N(u) có tương tự với U, nên các đánh giá về Item i của N(u) phần nào đó sẽ tương giống với đánh giá của U về Item i. Vì những người dùng láng giềng này có độ tương tự với U là khác nhau, do đó, ảnh hưởng của nó tới đánh giá của U là khác nhau. Do đó, ta sử dụng độ tương tự như trọng số khi tính giá trị trung bình của các đánh giá trên N(U).
- Ta sẽ chọn ra giá trị I sao cho P(U, I) là lớn nhất để tư vấn cho user U.

Câu 15:

Sơ đồ khối

Giải thích

- Cho bài toán gồm có tập người dùng U và tập Item I. Đồng thời xây dựng ma trận hữu ích P, có một vài người dùng đã đánh giá Item i, tuy nhiên cũng có nhiều ô chưa được hoàn thiện, cần phải dự đoán đánh giá của U với các Item mà nó chưa dự đoán.
- Ta sẽ tính toán độ tương tự giữa các Item qua công thức:

$$sim\ (i,j) = CV(i,j) = \cos(p_i, p_j) = \frac{\sum_{t \in S_{ij}} p_{ti} \times p_{tj}}{\sqrt{\sum_{t \in S_{ij}} p_{ti}^2 \times \sum_{t \in S_{ij}} p_{tj}^2}}$$

- Trong đó Sij là tập các user đã đánh giá cả hai item i và j.
- Sau đó, để dự đoán đánh giá của User U với Item i, thì ta sẽ chọn ra Qi(u) là tập top k item thuộc S(u) mà tương tự với i nhất. Sau đó, tính toán độ hữu ích của item i với user với u qua công thức:

$$p_{u,i} = \frac{\sum_{j \in Q(i)} sim\,(p_i, p_j) \times p_{u,j}}{\sum_{j \in Q(i)} sim\,(p_i, p_j)}$$

- Khi cần tư vấn cho user u thì chọn ra giá trị i lớn nhất sao cho Pui đạt max.

Câu 16:

Sơ đồ khối

Giải thích

- Nhận vào dữ liệu dạng nội dung, mô tả các feature của người dùng và tập mục.
- Giả sử mối quan tâm của người này ít liên quan tới người khác. Mối quan tâm của người sẽ theo các tính chất của mục.
- Ban đầu, ta lập thành phần phân tích nội dung để tiền xử lý các thuộc tính của mục và truy xuất ra các đặc trưng của mục. Qua đó biểu diễn được các mục dưới dạng dữ liệu có cấu trúc.
- Tiếp theo, sử dụng dữ liệu về các mục và phản hồi của người dùng, ta sẽ sử dụng mô hình học máy để tổng quát hoá thành mô hình sở thích của người dùng

- Xây dựng thành phần lọc để đối chiếu biểu diễn mục tiềm năng với sở thích của người dùng. Sử dụng các độ đo để chọn ra được các mục có độ liên quan lớn nhất với sở thích của người dùng.

Câu 17:

Đánh giá hiệu năng của hệ tư vấn

- Sử dụng người dùng nghiên cứu: Huy động tập người dùng thử và đánh giá, dữ liệu tương tác trực tiếp giữa người dùng và hệ thống. Tuy nhiên gặp nhiều hạn chế trong việc tuyển dụng người dùng.
- Trực tuyến:
  - Chọn người dùng thực làm việc với hệ thống.
  - Độ đo tỷ lệ chuyển đổi, đo tần suất người dùng chọn mục do hệ thống đề xuất.
  - Chọn 1 trong 2 thuật toán: Kiểm thử A/B, chọn ngẫu nhiên 2 nhóm người dùng A và B, A sử dụng 1 thuật toán, B sử dụng 1 thuật toán, như nhau về điều kiện và về cùng khoảng thời gian.
- Ngoại tuyến:
  - Sử dụng bộ dữ liệu lịch sử cho đánh giá: Netflix Prize.
  - Lợi thế: Có sẵn khung và độ đo đánh giá chuẩn.
  - Hạn chế: Dữ liệu quá khứ + hiện tại không phản ánh xu thế sau này.
  - Chấp nhận rộng rãi và là phương pháp phổ biến nhất.

Độ đo

$$precision@k(u) = \frac{|predicted_k(u) \cap relevant(u)|}{k}$$

$$recall@k(u) = \frac{|predicted_k(u) \cap relevant(u)|}{|relevant(u)|}$$

$$precision@(u) = \frac{|predicted(u) \cap relevant(u)|}{predicted(u)}$$

$$recall@(u) = \frac{|predicted(u) \cap relevant(u)|}{|relevant(u)|}$$

Hướng phân lớp

$$MSE = \frac{\sum_{(u,j) \in E} e_{uj}^2}{|E|}$$

$$RMSE = \sqrt{\frac{\sum_{(u,j) \in E} e_{uj}^2}{|E|}}$$

$$NRMSE = \frac{RMSE}{p_{max} - p_{min}}$$

$$MAE = \frac{\sum_{(u,j) \in E} |e_{uj}|}{|E|}$$

$$NMAE = \frac{MAE}{p_{max} - p_{min}}$$

Hướng hồi quy

Câu 18:

- Biểu diễn các ảnh trên mặt phẳng 2 chiều do có hai thuộc tính.
- Xét điểm dữ liệu mới (x, y). Tính khoảng cách từ điểm này tới tất cả các điểm khác trong bộ dữ liệu.
- Công thức tính khoảng cách là công thức khoảng cách Euclid.
- Lấy ra 3 examples gần nhất với điểm dữ liệu mới.
- Xét các nhãn của 3 examples.
- Nếu như có nhiều hơn 2 nhãn là 1, thì gán nhãn của điểm dữ liệu mới là 1.
- Ngược lại gán nhãn của điểm dữ liệu mới là 0.

Câu 19:

- Ban đầu, ta chọn ra F[1] là tập các tập mục phổ biến có độ dài 1.
- Ta thấy {a} và {c} có 6 lần xuất hiện ⇒ sup(a) > 0.5, sup(c) > 0.5.
- Ta thấy {b} có 7 lần xuất hiện ⇒ sup(b) > 0.5
- Ta thấy {d} và {e} có 2 lần xuất hiện trong CSGD ⇒ sup(d) < 0.5, sup(e) < 0.5

⇒ F[1] = {(a), (b), (c)}

- Ta lọc ra các ứng viên cho F[2]:
  - Xét Pk = (a), Qk = (b) ⇒ c(k+1) = (a, b)
  - Xét Pk = (a), Qk = (c) ⇒ c(k+1) = (a, c)
  - Xét Pk = (b), Qk = (c) ⇒ c(k+1) = (b, c)

- Cắt tỉa: Các con của các ứng viên trên đều thuộc F[1] nên không cần cắt bỏ.

⇒ C[2] ={(a, b), (b, c), (a, c)}.

- Xét (a, b), có 4 lần xuất hiện ⇒ sup(a, b) < 0.5
- Xét (a, c), có 4 lần xuất hiện ⇒ sup(a, c) < 0.5
- Xét (b, c), có 4 lần xuất hiện ⇒ sup(b, c) < 0.5

⇒ F[2] rỗng.

⇒ Không có tập phổ biến độ dài 2.

Câu 20:

Ta có ma trận nhầm lẫn:

TP: 12.

FP: 1.

FN: 3.

TN: 26.

Precision

P = TP / (TP + FP) = 12/13 = 0.923.

Recall

R = TP / (TP + FN) = 12/15 = 0.8

F1 Score

F1 = 2/((1/R) + (1/P)) = 2 / (13/12 + 15/12) = 0.857

Accuracy

A = (TP + TN) / (TP + TN + FP + FN) = 0.904

Error Rate

E = (FP + FN) / (TP + TN + FP + FN) = 0.095

Như vậy qua Cặp độ đo Precision và Recall, cùng với F1 score, thì ta thấy độ chính xác lên tới 85,7%.

Tuy nhiên khi đó với Accuracy và Error rate, thì có độ chính xác là 90,4%.