**MỤC LỤC**

[MỤC LỤC 1](#_Toc309304769)

[DANH MỤC HÌNH ẢNH 3](#_Toc309304770)

[DANH MỤC BẢNG 4](#_Toc309304771)

[DANH MỤC BIỂU ĐỒ 4](#_Toc309304772)

[1 Nhập đề 5](#_Toc309304773)

[2 Giới thiệu bài toán 5](#_Toc309304774)

[2.1 Giới thiệu bài toán 5](#_Toc309304775)

[2.1.1 Bài toán 1: Hỗ trợ người dùng viết CV theo ngành nghề. 5](#_Toc309304776)

[2.1.2 Bài toán 2: Hỗ trợ nhà tuyển dụng trong việc đăng tải yêu cầu công việc một cách chi tiết và có trọng số theo ngành nghề. 5](#_Toc309304777)

[2.1.3 Bài toán 3: Matching tool 6](#_Toc309304778)

[2.2 Kết quả mong muốn 6](#_Toc309304779)

[3 Giải quyết bài toán 6](#_Toc309304780)

[3.1 Cây quyết định 7](#_Toc309304781)

[3.1.1 Giới thiệu về cây quyết định 7](#_Toc309304782)

[3.1.2 Sử dụng cây quyết định để dự đoán lớp các dữ liệu chưa biết 8](#_Toc309304783)

[3.1.3 Một số độ đo thông dụng 9](#_Toc309304784)

[3.1.4 Nguyên tắc hoạt động của một số thuật toán xây dựng cây quyết định thường dùng 12](#_Toc309304785)

[3.1.5 So sánh một số thuật toán xây dựng cây quyết định 22](#_Toc309304786)

[3.1.6 Liên kết các cây quyết định 22](#_Toc309304787)

[3.1.7 Đánh giá cây quyết định trong lĩnh vực khai thác dữ liệu 23](#_Toc309304788)

[3.1.8 Ứng dụng cây quyết định vào bài toán 24](#_Toc309304789)

[3.2 Taxonomy 24](#_Toc309304790)

[3.3 Phương pháp phân loại động 24](#_Toc309304791)

[3.4 Semantic web 24](#_Toc309304792)

[4 Nền tảng lý thuyết kiến trúc cổng thông tin tìm việc 24](#_Toc309304793)

[4.1 Tại sao lại cần có kiến trúc cổng thông tin tìm việc Job Zoom 24](#_Toc309304794)

[4.2 Tại sao cần khai thác dữ liệu 25](#_Toc309304795)

[4.3 Giải quyết trường hợp dữ liệu không đủ để khai thác 25](#_Toc309304796)

[4.4 Tại sao cần cây đa cấp 25](#_Toc309304797)

[4.5 Tính khả thi của Job Zoom framework 25](#_Toc309304798)

[4.6 Một số cổng thông tin tìm việc hiện tại đã có 25](#_Toc309304799)

[5 Kiến trúc cổng thông tin tìm việc JobZoom 25](#_Toc309304800)

[5.1 Điều kiện ra đời 25](#_Toc309304801)

[5.2 Mô hình kiến trúc khắc phục những điểm yếu của các website tìm việc hiện có 25](#_Toc309304802)

[5.3 Kiến trúc framework 25](#_Toc309304803)

[5.3.1 Kiến trúc tổng quan framework 25](#_Toc309304804)

[5.3.2 Matching tool 25](#_Toc309304805)

[5.3.3 Data mining 25](#_Toc309304806)

[5.4 Mô hình kiến trúc Job Zoom có nhiều ưu điểm vượt trội 26](#_Toc309304807)

[5.4.1 Kiến trúc linh hoạt và khả năng mở rộng của framework 26](#_Toc309304808)

[5.4.2 Khả năng ứng dụng vào những ngành nghề khác nhau 26](#_Toc309304809)

[5.4.3 Khả năng ứng dụng vào những lĩnh vực khác 26](#_Toc309304810)

[5.4.4 Triển khai nhanh chóng 26](#_Toc309304811)

[5.4.5 Đảm bảo hiệu năng hệ thống 26](#_Toc309304812)

[6 Giải pháp công nghệ 26](#_Toc309304813)

[6.1 .NET Framework 26](#_Toc309304814)

[6.2 SQL Server Code Name “Denali” CTP3 27](#_Toc309304815)

[6.2.1 Analysis Service 27](#_Toc309304816)

[6.2.2 Database Engine 27](#_Toc309304817)

[6.2.3 MDX/DMX 27](#_Toc309304818)

[6.2.4 Pivot table 27](#_Toc309304819)

[6.3 jQuery 27](#_Toc309304820)

[6.4 MVC 27](#_Toc309304821)

[7 So sánh Job Zoom với các website xin việc hiện tại 27](#_Toc309304822)

[8 Đánh giá và hướng phát triển 27](#_Toc309304823)

[8.1 Những điểm làm được 27](#_Toc309304824)

[8.2 Những điểm hạn chế 27](#_Toc309304825)

[8.3 Hướng phát triển 27](#_Toc309304826)

[8.3.1 Xác thực độ tin cậy của CV 27](#_Toc309304827)

[8.3.2 Sematic web 28](#_Toc309304828)

[8.3.3 Phân cụm dữ liệu 28](#_Toc309304829)

[TÀI LIỆU THAM KHẢO 29](#_Toc309304830)

**DANH MỤC HÌNH ẢNH**

[Hình 1. Ví dụ về cây quyết định 8](#_Toc309296343)

[Hình 2. Phân chia bảng D thành những tập con Si 13](#_Toc309296344)

[Hình 3. Phân chia Si dựa vào phân lớp 14](#_Toc309296345)

[Hình 4. Độ lợi thông tin thuộc tính Travel cost/Km 15](#_Toc309296346)

[Hình 5. Độ lợi thông tin các thuộc tính còn lại trong Bảng 1 17](#_Toc309296347)

[Hình 6. Node gốc của cây quyết định sau lần lặp đầu tiên 18](#_Toc309296348)

[Hình 7. Bảng D được phân chia sau lần lặp đầu tiên 18](#_Toc309296349)

[Hình 8. Cây quyết định sau lần lặp đầu tiên 19](#_Toc309296350)

[Hình 9. Dữ liệu cho lần phân lớp thứ 2 19](#_Toc309296351)

[Hình 10. Tính Impurity degree cho lần phân lớp thứ 2 20](#_Toc309296352)

[Hình 11. Tính Impurity degree các thuộc tính cho lần phân lớp thứ 2 20](#_Toc309296353)

[Hình 12. Bảng dữ liệu sau khi chia theo phân lớp Gender 21](#_Toc309296354)

[Hình 13. Cây quyết định sau lần phân lớp thứ 2 21](#_Toc309296355)

[Hình 14. Cây quyết định đầy đủ sau 3 lần phân lớp 22](#_Toc309296356)

**DANH MỤC BẢNG**

[Bảng 1. Ví dụ bảng dữ liệu lựa chọn phương tiện di chuyển 7](#_Toc309296357)

[Bảng 2. Ví dụ về bảng dữ liệu cần dự đoán phương tiện di chuyển 1 8](#_Toc309296358)

[Bảng 3. Ví dụ về bảng dữ liệu cần dự đoán phương tiện di chuyển 2 9](#_Toc309296359)

[Bảng 4. Lợi ích khi chia bảng D theo thuộc tính “Travel cost/km” 16](#_Toc309296360)

[Bảng 5. Kết quả độ lợi thông tin sau khi phân chia bảng D theo từng thuộc tính 18](#_Toc309296361)

[Bảng 6. Bảng dữ liệu cho lần phân lớp thứ 3 22](#_Toc309296362)

**DANH MỤC BIỂU ĐỒ**

[Biểu đồ 1. Mối liên hệ giữa giá trị Entropy cực đại và số lượng phân lớp 11](#_Toc309296363)

[Biểu đồ 2. Mối liên hệ giữa giá trị Gini index cực đại và số lượng phân lớp 12](#_Toc309296364)

# Nhập đề

# Giới thiệu bài toán

## Giới thiệu bài toán

### Bài toán 1: Hỗ trợ người dùng viết CV theo ngành nghề.

**Bạn là người tìm việc?** Có thể bạn đã từng gặp khó khăn trong việc viết sơ yếu lý lịch (CV hay còn gọi là resume) để tạo ấn tượng cho nhà tuyển dụng, tiếp thị cho bản thân mình. Bên cạnh đó, mẫu CV sẽ khác nhau đối với từng ngành nghề khác nhau. Để lọt vào danh sách top ten ứng viên của nhà tuyển dụng, bạn cần phải có một sơ yếu lý lịch ngắn gọn, xúc tích nhưng cần rõ ràng và đầy đủ về quá trình làm việc, trình độ chuyên môn, học vấn hay các kỹ năng mềm, việc viết CV này thật sự trở nên khó khăn khi bạn vẫn loay hoay chưa biết phải trình bày hay liệt kê những thông tin như thế nào. Những ứng dụng tìm việc hiện tại không thể gợi ý cho bạn tạo một CV chuẩn cho một ngành nghề cụ thể hay nói khác đi là không hỗ trợ người dùng tạo CV xin việc đầy đủ và rõ ràng đối với ngành nghề mà người xin việc muốn apply.

Khi người tìm việc muốn tìm một công việc nào đó, bên cạnh những thông tin cá nhân, ứng dụng sẽ gợi ý một CV chuẩn về ngành nghề, người dùng sẽ tạo ra sơ yếu lý lịch cho mình dựa vào các thuộc tính chương trình gợi ý. Ngoài ra, khi cần apply vào một ví trí của một nhà tuyển dụng nào, ứng dụng sẽ gợi ý người dùng những thông tin cần bổ sung cho vị trí tìm việc.

### Bài toán 2: Hỗ trợ nhà tuyển dụng trong việc đăng tải yêu cầu công việc một cách chi tiết và có trọng số theo ngành nghề.

**Bạn là nhà tuyển dụng?** Các website tìm việc hiện nay, đều cho nhà tuyển dụng đăng thông tin dưới dạng văn bản thô, không đưa ra gợi ý cho bạn về các ứng viên có khả năng đáp ứng yêu cầu công việc dựa vào các trọng số các yêu cầu của bạn đưa ra. Nhà tuyển dụng khi muốn đăng thông tin tuyển dụng, các website xin việc hiện tại không thể đánh giá mức độ đáp ứng của những ứng viên dựa vào những thông tin nhà tuyển dụng cung cấp, không thể biết ứng viên đáp ứng được bao nhiêu phần trăm những yêu cầu này, nhà tuyển dụng phải tự đánh giá bằng việc đọc những CV của ứng viên sau đó lựa chọn, việc này gây mất nhiều thời gian và kinh phí.

Nhà tuyển dụng có thể đánh giá trọng số các thông tin yêu cầu về công việc. Ứng dụng đánh giá mức độ phù hợp của ứng viên và liệt kê những ứng viên “tiềm năng” dựa vào thông tin của nhà tuyển dụng cung cấp.

### Bài toán 3: Matching tool

Đối với người tìm việc: Matching tool gợi ý người dùng hoàn thiện, bổ sung những thiếu sót của CV, tăng cơ hội apply vào một vị trí tại một công ty cụ thể.

Đối với nhà tuyển dụng: Matching tool đánh giá ứng viên xin apply vào công việc, hỗ trợ cho nhà tuyển dụng trong việc đánh giá ứng viên

## Kết quả mong muốn

# Giải quyết bài toán

Để giải quyết bài toán trên, nhóm chúng tôi đưa ra giải pháp như sau:

Áp dụng lý thuyết khai thác dữ liệu sử dụng cây quyết định, từ đó tạo ra cây thuộc tính những yêu cầu công việc chung nhất ứng với từng ngành nghề cụ thể. Cây quyết định này sẽ được sử dụng trong việc giải quyết bài toán 1, nhằm gợi ý người dùng viết một CV đơn giản, phù hợp với công việc hiện tại mà người xin việc mong muốn apply. Để đáp ứng điều này, nhóm cần phải phân loại dữ liệu theo ngành nghề và vị trí làm việc, nói khác hơn chính là việc liên kết các cây quyết định lại với nhau nhằm tạo ra cây quyết định lớn, đảm bảo trong việc khai thác dữ liệu từ cây để gợi ý cho người xin việc một mẫu CV cơ bản đối với từng ngành nghề cụ thể. Ngoài ra, đối với người tìm việc đã liệt kê những kỹ năng kinh nghiệm làm việc của mình mà chưa biết phải apply vào công việc nào, việc áp dụng cây quyết định này cũng có thể đáp ứng được việc gợi ý công việc cho người dùng apply

(Hình cây quyết định)

Đối với bài toán 2, nhóm chúng tôi sử dụng taxonomy để đăng tải CV cũng như yêu cầu công việc của người dùng và nhà tuyển dụng dưới dạng cây. Việc so sánh những cây CV người xin việc cung cấp và cây yêu cầu công việc của nhà tuyển dụng, chương trình sẽ đánh giá mức độ đáp ứng của ứng viên dựa vào việc so sánh các cây dữ liệu này. Ngoài ra, taxonomy còn được sử dụng trong việc gợi ý cho người dùng bổ sung những thông tin còn thiếu để hoàn thiện CV khi xin apply vào một vị trí làm việc nào đó tại công ty tuyển dụng.

(Hình so sánh 2 cây)

Việc ứng dụng cây quyết định và taxonomy lại phát sinh bài toán đảm bảo hiệu năng hệ thống khi ứng dụng cây quyết định và taxonomy…

## Cây quyết định

### Giới thiệu về cây quyết định

* Cây quyết định là một cây phân cấp có cấu trúc.
* Dùng để phân lớp đối tượng dựa vào dãy các luật (series of rules), các luật này được sinh ra từ tập dữ liệu (training set).
* Các thuộc tính phân lớp có kiểu dữ liệu là binary, nominal, ordinal, continuos.
* Ví dụ: Cho tập dữ liệu (training set) như sau:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Thuộc tính (Attribute) | | | | Thuộc tính phân lớp (Class) |
| Gender | Car ownership | Travel cost ($/km) | Income level | Transportation mode |
| Male | 0 | Cheap | Low | Bus |
| Male | 1 | Cheap | Medium | Bus |
| Female | 1 | Cheap | Medium | Train |
| Female | 0 | Cheap | Low | Bus |
| Male | 1 | Cheap | Medium | Bus |
| Male | 0 | Standard | Medium | Train |
| Female | 1 | Standard | Medium | Train |
| Female | 1 | Expensive | High | Car |
| Male | 2 | Expensive | Medium | Car |
| Female | 2 | Expensive | High | Car |

Bảng 1. Ví dụ bảng dữ liệu lựa chọn phương tiện di chuyển

* Tập dữ liệu trên mô tả 10 đối tượng, mỗi đối tượng được miêu tả bằng 4 thuộc tính là “Gender” (kiểu dữ liệu binary), “Car ownership” (quantitative interger), “Travel cost/km” (ordinal), “Income level” (ordinal) và thuộc tính phân loại – category attribute – “Transportation mode” (ordinal).
* Từ tập dữ liệu trên, chúng ta có thể tạo ra cây quyết định như sau:



Hình 1. Ví dụ về cây quyết định

* Trong cây quyết định trên, thuộc tính “Income level” không xuất hiện trong cây; vì dựa vào Bảng 1 (trang 7), thuộc tính “Travel cost/Km” sẽ sinh ra cây quyết định dùng để phân lớp tốt hơn “Income level”

### Sử dụng cây quyết định để dự đoán lớp các dữ liệu chưa biết

* Mục đích chính của cây quyết định là dùng để xác định lớp hay nói khác đi là dự đoán lớp của các dữ liệu chưa biết dựa vào cây quyết định được sinh ra từ tập dữ liệu (training data)
* Ví dụ: dựa vào ví dụ ở phần 3.1.1 Giới thiệu về cây quyết định (trang 7). Cho tập dữ liệu cần dự đoán sau đây:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Gender | Car ownership | Travel cost ($/km) | Income level | Transportation mode |
| Ngoc | Female | 1 | Cheap | High | ? |
| Hieu | Male | 0 | Standard | High | ? |
| Phuc | Male | 1 | Cheap | Medium | ? |
| Nguyen | Male | 2 | Expensive | High | ? |

Bảng 2. Ví dụ về bảng dữ liệu cần dự đoán phương tiện di chuyển 1

* Dựa vào cây quyết định (Hình 1, trang 8), cây quyết định sẽ được duyệt từ nút gốc “Travel cost/km”, dãy các luật sau sẽ được sinh ra:
* Nếu “Travel cost/Km” là **Expensive** thì người đó sẽ chọn phương tiện di chuyển là **car**.
* Nếu “Travel cost/Km” là **Standard** thì người đó sẽ chọn phương tiện di chuyển là **train**
* Nếu “Travel cost/Km” là **Cheap**, chúng ta sẽ xem xét thuộc tính “Gender”:
  + Nếu “Gender” là **Male**, người đó sẽ chọn phương tiện là **bus**
  + Nếu “Gender là **Female**, thì xem người đó sở hữu bao nhiêu xe hơi (thuộc tính “Car ownership”). Nếu số xe sở hữu là **0**, thì người đó sẽ chọn phương tiện di chuyển là **bus**; ngược lại, nếu số xe sở hữu lớn hơn hay bằng **1**, thì người đó sẽ chọn phương tiện di chuyển là **train**.
* Bảng 2 sẽ được dự đoán như sau:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Gender | Car ownership | Travel cost ($/km) | Income level | Transportation mode |
| Ngoc | Female | 1 | Cheap | High | Train |
| Hieu | Male | 0 | Standard | High | Train |
| Phuc | Male | 1 | Cheap | Medium | Bus |
| Nguyen | Male | 2 | Expensive | High | Car |

Bảng 3. Ví dụ về bảng dữ liệu cần dự đoán phương tiện di chuyển 2

* **Một số chú ý khi sử dụng cây quyết định:**
* Phụ thuộc rất nhiều vào training data, tập dữ liệu training data càng lớn thì cây quyết định sẽ đáng tin cậy hơn.
* Không thể nói cây quyết định được sinh ra từ cây quyết định trên là tập luật tốt nhất
* Có nhiều thuật toán phân lớp: ID3, J48, C4.5, C5, CART (Classification and Regression Tree), … Việc lựa chọn thuật toán phụ thuốc vào rất nhiều yếu tố, trong đó yếu tố cấu trúc dữ liệu ảnh hưởng rất nhiều đến kết quả của thuật toán. Chẳn hạn, thuật toán ID3 và CART hiệu quả cho việc phân lớp đối với các dữ liệu số (quantitative value), trong khi đó, thuật toán J48, C4.5 có hiệu quả hơn đối với dữ liệu Qualititive value (ordinal, binary, nominal)

### Một số độ đo thông dụng

* Cho bảng dữ liệu bao gồm các thuộc tính và thuộc tính phân lớp, chúng ta có thể đo được tính đồng nhất hay không đồng nhất thông qua thuộc tính phân lớp. Bảng dữ liệu có tính đồng nhất nếu nó có duy nhất một phân lớp. Ngược lại, nếu nó có nhiều phân lớp khác nhau, thì bảng dữ liệu có tính không đồng nhất hay tính pha trộn. Chúng ta có thể đo được mức độ pha trộn (Impurity Degree); Entropy, độ đo Gini và classification error là những cách tính mức độ pha trộn thông dụng nhất.

*Pj: xác suất xảy ra phân lớp j*

* Ví dụ: Chúng ta cùng nhìn lại Bảng 1. Ví dụ bảng dữ liệu lựa chọn phương tiện di chuyển. “Transportation mode” có 3 nhóm Bus, Car và Train. Bảng 1 có 10 dòng dữ liệu, trong đó “Transportation mode” có 4 buses, 3 cars, 3 trains (4B, 3C, 3T).
* Xác suất để xảy ra cho từng phân lớp là:

#### Entropy

Khi bảng dữ liệu có duy nhất 1 phân lớp thì Entropy sẽ bằng 0 vì xác suất là 1 và . Entropy sẽ đạt giá trị cực đại khi tất cả thuộc tính phân lớp có xác suất xảy ra bằng nhau. Biểu đồ dưới đây thể hiện giá trị cực đại của Entropy sẽ thay đổi phụ thuộc vào số lượng thuộc tính phân lớp n, trong trường hợp xác xuất tất cả thuộc tính phân lớp thì . Giá trị của Entropy sẽ lớn hơn 1 khi số lượng thuộc tính phân lớp nhiều hơn 2



Biểu đồ 1. Mối liên hệ giữa giá trị Entropy cực đại và số lượng phân lớp

#### Gini index

* Một cách khác để tính “Impurity degree”.
* Chúng ta cùng quay lại ví dụ ở trên
* Khi bảng dữ liệu có duy nhất 1 phân lớp thì Gini index sẽ bằng 0 vì xác suất bằng 1 và . Cũng giống như Entropy, Gini index sẽ đạt giá trí cực đại khi tất cả thuộc tính phân lớp có xác suất xảy ra bằng nhau. Biểu đồ dưới đây thể hiển giá trị cực đại của Gini index sẽ khác nhau phụ thuộc vào số lượng phân lớp n, khi xác xuất tất cả thuộc tính phân lớp



Biểu đồ 2. Mối liên hệ giữa giá trị Gini index cực đại và số lượng phân lớp

#### Classification error

* Trong ví dụ trên
* Cũng giống như Entropy và Gini index, Classification error sẽ bằng 0 khi bảng dữ liệu có duy nhất 1 phân lớp vì xác suất bằng 1 và .

### Nguyên tắc hoạt động của một số thuật toán xây dựng cây quyết định thường dùng

* Các thuật toán xây dựng cây quyết định thường được sử dụng nhất là ID3, C4.5 và CART (classification and regression trees).
* Nhìn chung, các thuật toán xây dựng cây quyết định đều được xây dựng trên quy tắc đệ quy. Ví dụ, Hunt là thuật toán dựng cây quyết định, đệ quy theo nút của cây, bắt đầu từ nút gốc. Mặc dù kết quả đạt được từ thuật toán Hunt không được tối ưu, nhưng đây là một trong những một trong những thuật toán dùng để xây dựng cây quyết định sớm nhất.

#### Lần lặp đầu tiên

* Giả sử, chúng ta có một bảng dữ liệu chứa các thuộc tính và thuộc tính phân lớp, tạm gọi bảng này là D. Từ bảng D, chúng ta sẽ lấy ra từng cột thuộc tính trong bảng để đối chiếu với các giá trị của thuộc tính phân lớp. Nếu chúng ta có p cột dữ liệu, và lấy từng phần tử p là tập con của D, tạm gọi từng tập hợp này là Si. Bảng D là tập hợp những Si và thuộc tính phân lớp.



Hình 2. Phân chia bảng D thành những tập con Si

* Sau khi phân chia bảng dữ liệu D như trên, chúng ta sẽ tính mức độ pha trộn (tham khảo cách tính tại phần 3.1.3 Một số độ đo thông dụng)
* Ví dụ, dựa vào Bảng 1. Ví dụ bảng dữ liệu lựa chọn phương tiện di chuyển. chúng ta có thể tính “Impurity degree” dựa vào thuộc tính phân lớp “Transportation mode”. “Transportation mode” có 4 busses, 3 cars và 3 trains (4B, 3C, 3T)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Thuộc tính (Attribute) | | | | Thuộc tính phân lớp (Class) |
| Gender | Car ownership | Travel cost ($/km) | Income level | Transportation mode |
| Male | 0 | Cheap | Low | Bus |
| Male | 1 | Cheap | Medium | Bus |
| Female | 1 | Cheap | Medium | Train |
| Female | 0 | Cheap | Low | Bus |
| Male | 1 | Cheap | Medium | Bus |
| Male | 0 | Standard | Medium | Train |
| Female | 1 | Standard | Medium | Train |
| Female | 1 | Expensive | High | Car |
| Male | 2 | Expensive | Medium | Car |
| Female | 2 | Expensive | High | Car |

4B, 3C, 3T

Entropy 1.571

Gini index 0.660

Classification index 0.600

* Sau khi phân chia bảng D thành từng tập con Si, chúng ta tiếp tục phân chia Si để tính “Impurity degree” cho từng giá trị thuộc tính trong từng tập Si



Hình 3. Phân chia Si dựa vào phân lớp

* Tiếp tục ví dụ trên, thuộc tính “Travel cost/km” có 3 giá trị, đó là “Cheap”, “Standard” và “Expensive”.



Hình 4. Độ lợi thông tin thuộc tính Travel cost/Km

* **Độ lợi thông tin (Information Gain):**
  + Lý do có sự khác nhau giữa các cách tính mức độ pha trộn “Impurity degree” giữa bảng D và các tập con Si bởi vì chúng ta sẽ tiến hành so sánh “Impurity degree” trước và sau khi phân chia bảng thành những tập con (ví dụ như phân chia D thành từng tập con Si). Information gain là một phương pháp để đo lường sự khác nhau này. Chúng ta sẽ đạt lợi ích gì khi chia bảng dữ liệu dựa vào một số giá trị của thuộc tính
* Xét ví dụ trên, bảng D có 3 phân lớp 4B, 3C và 3T và có Entropy là 1.571. Bây giờ chúng ta sẽ sử dụng thuộc tính “Travel cost/km” để chia bảng thành 3 phần:
* “Travel cost/km” có giá trị là Cheap, thuộc tính phân lớp có 4B và 1T

* “Travel cost/km” có giá trị là Standard, thuộc tính phân lớp có 2T, vì lúc này chỉ có duy nhất 1 thuộc tính phân lớp)
* “Travel cost/km” có giá trị là Expensive, thuộc tính phân lớp có 3C, vì lúc này chỉ có duy nhất 1 thuộc tính phân lớp)
* Tương tự chúng ta sẽ tính được độ đo Gini index và Classification error:

|  |  |
| --- | --- |
| **Lợi ích khi chia theo thuộc tính “Travel cost/km” dựa vào** | |
| Entropy | 1.210 |
| Gini index | 0.500 |
| Classification error | 0.500 |

Bảng 4. Lợi ích khi chia bảng D theo thuộc tính “Travel cost/km”

* Chúng ta thực hiện lần lượt cho các thuộc tính còn lại của bảng D: “Gender”, “Car ownership” và “Income level”



Hình 5. Độ lợi thông tin các thuộc tính còn lại trong Bảng 1

* Bảng dưới đây sẽ cho chúng ta thấy độ lợi thông tin cho tất cả 4 thuộc tính trong bảng D. Chúng ta không cần tính “Impurity degree” dựa trên cả 3 độ đo Entropy, Gini index và Classification error, chỉ cần chọn lựa 1 trong 3 độ đo trên.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Kết quả của lần lặp đầu tiên** | | | | |
| **Độ lợi** | **Gender** | **Car ownership** | **Travel cost/Km** | **Income level** |
| Entropy | 0.125 | 0.534 | **1.210** | 0.695 |
| Gini index | 0.060 | 0.207 | **0.500** | 0.293 |
| Classification error | 0.100 | 0.200 | **0.500** | 0.300 |

Bảng 5. Kết quả độ lợi thông tin sau khi phân chia bảng D theo từng thuộc tính

* Sau khi tính được độ lợi thông tin cho từng thuộc tính, chúng ta sẽ lựa chọn thuộc tính có độ lợi thông tin cao nhất để làm node gốc cho cây quyết định.



Hình 6. Node gốc của cây quyết định sau lần lặp đầu tiên

* Bảng D sẽ được phân chia theo thuộc tính “Travel cost/km” như sau:



Hình 7. Bảng D được phân chia sau lần lặp đầu tiên



Hình 8. Cây quyết định sau lần lặp đầu tiên

#### Những lần phân lớp tiếp theo

Trong phần này sẽ sử dụng lại ví dụ ở phần 3.1.4.1

##### Lần phân lớp thứ 2

* Sau lần lặp đầu tiên, chúng ta cần cập nhật lại bảng dữ liệu. Khi chúng ta phân chia bảng D theo node gốc “Travel cost/km”, những dòng dữ liệu có giá trị thuộc tính “Travel cost/km” là Expensive và Standard đã là phân lớp thuần khiết, chúng ta không cần sử dụng lại những dòng dữ liệu này để phân lớp. Trong lần lặp này, “Travel cost/km” chỉ còn lại giá trị Cheap, chúng ta sẽ loại bỏ thuộc tính này



Hình 9. Dữ liệu cho lần phân lớp thứ 2

* Tính “Impurity degree” cho bảng chính



Hình 10. Tính Impurity degree cho lần phân lớp thứ 2

* Tính “Impurity degree” cho các thuộc tính của bảng chính



Hình 11. Tính Impurity degree các thuộc tính cho lần phân lớp thứ 2

* Độ lợi thông tin khi phân lớp theo thuộc tính “Gender” lớn nhất. Chúng ta sẽ phân chia bảng dữ liệu như sau:



Hình 12. Bảng dữ liệu sau khi chia theo phân lớp Gender

* Cây quyết định sau lần phân lớp thứ 2:



Hình 13. Cây quyết định sau lần phân lớp thứ 2

##### Lần phân lớp thứ 3

Tương tự, sau lần phân lớp thứ 2, bảng dữ liệu cho lần phân lớp thứ 3 như sau:



Bảng 6. Bảng dữ liệu cho lần phân lớp thứ 3

Nếu chia bảng trên theo “Car ownership” hay “Income level” thì bảng dữ liệu được chia đều có duy nhất 1 phân lớp. Chúng ta có thể chia bảng dữ liệu theo 1 trong 2 thuộc tính này.



Hình 14. Cây quyết định đầy đủ sau 3 lần phân lớp

### So sánh một số thuật toán xây dựng cây quyết định

### Liên kết các cây quyết định

### Đánh giá cây quyết định trong lĩnh vực khai thác dữ liệu

#### Điểm mạnh

* **Cây quyết định sinh ra các quy tắc hiểu được**

Cây quyết định có thể sinh ra các quy tắc được chuyển đổi sang tiếng Anh hay câu lệnh SQL. Thậm chí đối với những tập dữ liệu lớn làm cho hình dáng cây quyết định lớn và phức tạp, việc duyệt cây cũng rất dễ dàng. Bất cứ một sự phân lớp hay dự đoán nào đều tương đối minh bạch. Đây là ưu điểm nổi bật của cây quyết định.

* **Cây quyết định có thể thực thi trong những lĩnh vực hướng quy tắc**

Cây quyết định là sự lựa chọn hoàn hảo cho những lĩnh vực có quy tắc, từ lĩnh vực di truyền đến các những quá trình công nghiệp chứa các quy tắc ẩn, không rõ ràng (underlying rules), dữ liệu lỗi chưa được tiền xử lý khá phức tạp và tối nghĩa. Cây quyết định là một lựa chọn tối ưu khi cần tìm ra những quy tắc ẩn, không rõ ràng trong kho dữ liệu (data warehouse)

* **Dễ dàng tính toán trong khi phân lớp**

Những thuật toán xây dựng cây quyết định thường tạo ra cây với số phân nhánh thấp và kiểm tra đơn giản tại từng node. Những thuật toán này thường kiểm tra bằng cách so sánh số, xem xét phần tử của một tập hợp hay các phép nối đơn giản, những thao tác này sẽ được chuyển thành các toán hàm logic và số nguyên, đây là những toán hạng thực thi nhanh và chi phí tối ưu. Trong môi trường thương mại, các mô hình dự đoán này được sử dụng để phân lớp hàng triệu, thậm chí hàng tỷ bản ghi, bởi vậy có thể nói đây là một ưu điểm quan trọng của cây quyết định.

* **Cây quyết định xử lý với cả thuộc tính liên tục và thuộc tính rời rạc**

Các thuộc tính liên tục hay rời rạc đều có thể xử lý bằng cây quyết định. Tuy nhiên, thuộc tính liên tục cần nhiều tài nguyên tính toán hơn, được phân chia bằng việc chọn ra một ngưỡng trong tập các giá trị đã được sắp xếp của thuộc tính đó.

* **Thể hiện rõ ràng những thuộc tính quan trọng nhất cho việc dự đoán phân lớp**

Các thuật toán xây dựng cây quyết định chỉ ra những thuộc tính dùng để phân chia tốt nhất tập dữ liệu đào tạo bắt đầu từ node gốc của cây. Chính vì vậy, chúng ta có thể thấy được thuộc tính nào là quan trọng, có mức độ phụ thuộc lớn cho việc dự đoán hay phân lớp

#### Điểm yếu

Mặc dù có những điểm mạnh nổi bật trên, cây quyết định vẫn không tránh khỏi những điểm yếu. Cây quyết định không thích hợp lắm với những bài toán mực tiêu, như là dự đoán giá trị của thuộc tính liên tục: thu nhập, huyết áp, lãi suất ngân hàng… Bên cạnh đó, cây quyết định cũng khó giải quyết với những dữ liệu thời gian liên tục.

* **Cây quyết định dễ xảy ra lỗi khi có nhiều phân lớp**

Một số thuật toán chỉ tao tác với những lớp giá trị nhị phân dạng “có/không” hay “đồng ý/từ chối”. Số khác có thể chỉ định các bản ghi vào một số lớp bất kỳ nhưng dễ xảy ra lỗi khi ứng với một phân lớp, số lượng dữ liệu đào tạo nhỏ.

* **Chi phí tính toán để đào tạo cao**

Điểm này nghe có vẻ mâu thuẩn với điểm mạnh phía trên. Quá trình xây dựng cây quyết định khác đắt về mặt tính toán. Điều này cũng dễ hiểu vì cây quyết định có nhiều node trong trước khi đi đến node lá cuối cùng, ứng với từng node, chúng ta cần tính một độ đo (hay tiêu chuẩn phân chia) trên từng thuộc tính. Bên cạnh đó, đối với thuộc tính liên tục, chúng ta còn phải thêm thao tác sắp xếp lại dữ liệu theo thứ tự giá trị của thuộc tính đó. Quá trình này chọn ra những thuộc tính để phân lớp tốt nhất. Một số thuật toán còn sử dụng tổ hợp các thuộc tính kết hợp với nhau có trọng số để phát triển cây. Quá trình cắt tỉa cây chi phí cũng khá cao vì trong quá trình cắt tỉa, nhiều cây con sẽ được tạo ra và so sánh.

### Ứng dụng cây quyết định vào bài toán

## Taxonomy

+Phân loại tag

## Phương pháp phân loại động

## Semantic web

Tạo cây có mối quan hệ giữa các node

# Nền tảng lý thuyết kiến trúc cổng thông tin tìm việc

## Tại sao lại cần có kiến trúc cổng thông tin tìm việc Job Zoom

Chứng minh số lượng người tìm việc trực tuyến đang tăng cao?

Chứng minh người dùng cần hỗ trợ trong việc viết cv?

Chứng minh doanh nghiệp cần hỗ trợ đánh giá ứng viên?

Tại sao lại phát triển dưới hình thức framework mà không triển khai dưới dạng ứng dụng?

## Tại sao cần khai thác dữ liệu

## Giải quyết trường hợp dữ liệu không đủ để khai thác

## Tại sao cần cây đa cấp

## Tính khả thi của Job Zoom framework

## Một số cổng thông tin tìm việc hiện tại đã có

# Kiến trúc cổng thông tin tìm việc JobZoom

## Điều kiện ra đời

## Mô hình kiến trúc khắc phục những điểm yếu của các website tìm việc hiện có

Job Zoom framework giải quyết những điểm yếu nào của các website tìm việc hiện tại

## Kiến trúc framework

### Kiến trúc tổng quan framework

### Matching tool

+

+ Cách tính điểm

### Data mining

+ Decision tree

+ Đảm bảo hiệu năng hệ thống.

Hệ thống tự động tạo view, khai thác dữ liệu và lưu dữ liệu vào kho để truy xuất nhanh chóng và dễ dàng. Quá trình này được thực hiện độc lập trên một máy chủ khác …

## Mô hình kiến trúc Job Zoom có nhiều ưu điểm vượt trội

### Kiến trúc linh hoạt và khả năng mở rộng của framework

#### Interface minh chứng CV

#### Interface semantic web

#### Interface data mining

Tuỳ thuộc vào điều kiện sẽ có cách mining khác nhau

### Khả năng ứng dụng vào những ngành nghề khác nhau

### Khả năng ứng dụng vào những lĩnh vực khác

### Triển khai nhanh chóng

Người dùng không cần thay đổi cấu trúc bảng trong database.

Cách triển khai: …

### Đảm bảo hiệu năng hệ thống

# Giải pháp công nghệ

## .NET Framework

## SQL Server Code Name “Denali” CTP3

### Analysis Service

### Database Engine

### MDX/DMX

### Pivot table

## jQuery

## MVC

# So sánh Job Zoom với các website xin việc hiện tại

# Đánh giá và hướng phát triển

## Những điểm làm được

## Những điểm hạn chế

## Hướng phát triển

### Xác thực độ tin cậy của CV

### Sematic web

### Phân cụm dữ liệu

**TÀI LIỆU THAM KHẢO**