**MỤC LỤC**

[MỤC LỤC 1](#_Toc309314334)

[DANH MỤC HÌNH ẢNH 3](#_Toc309314335)

[DANH MỤC BẢNG 4](#_Toc309314336)

[DANH MỤC BIỂU ĐỒ 4](#_Toc309314337)

[1 Nhập đề 5](#_Toc309314338)

[2 Giới thiệu bài toán 6](#_Toc309314339)

[2.1 Giới thiệu bài toán 6](#_Toc309314340)

[2.1.1 Bài toán 1: Hỗ trợ người dùng viết CV theo ngành nghề. 6](#_Toc309314341)

[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. 6](#_Toc309314342)

[2.1.3 Bài toán 3: Matching tool 7](#_Toc309314343)

[2.2 Kết quả mong muốn 7](#_Toc309314344)

[3 Giải quyết bài toán 7](#_Toc309314345)

[3.1 Cây quyết định 8](#_Toc309314346)

[3.1.1 Giới thiệu về cây quyết định 8](#_Toc309314347)

[3.1.2 Sử dụng cây quyết định để dự đoán lớp các dữ liệu chưa biết 9](#_Toc309314348)

[3.1.3 Một số độ đo thông dụng 10](#_Toc309314349)

[3.1.4 Nguyên tắc hoạt động của một số thuật toán xây dựng cây quyết định thường dùng 13](#_Toc309314350)

[3.1.5 So sánh một số thuật toán xây dựng cây quyết định 23](#_Toc309314351)

[3.1.6 Liên kết các cây quyết định 23](#_Toc309314352)

[3.1.7 Đánh giá cây quyết định trong lĩnh vực khai thác dữ liệu 24](#_Toc309314353)

[3.1.8 Ứng dụng cây quyết định vào bài toán 25](#_Toc309314354)

[3.2 Taxonomy 25](#_Toc309314355)

[3.3 Phương pháp phân loại động 25](#_Toc309314356)

[3.4 Semantic web 25](#_Toc309314357)

[4 Nền tảng lý thuyết kiến trúc cổng thông tin tìm việc 25](#_Toc309314358)

[4.1 Tại sao lại cần có kiến trúc cổng thông tin tìm việc Job Zoom 25](#_Toc309314359)

[4.2 Tại sao cần khai thác dữ liệu 26](#_Toc309314360)

[4.3 Giải quyết trường hợp dữ liệu không đủ để khai thác 26](#_Toc309314361)

[4.4 Tại sao cần cây đa cấp 26](#_Toc309314362)

[4.5 Tính khả thi của Job Zoom framework 26](#_Toc309314363)

[4.6 Một số cổng thông tin tìm việc hiện tại đã có 26](#_Toc309314364)

[5 Kiến trúc cổng thông tin tìm việc JobZoom 26](#_Toc309314365)

[5.1 Điều kiện ra đời 26](#_Toc309314366)

[5.2 Mô hình kiến trúc khắc phục những điểm yếu của các website tìm việc hiện có 26](#_Toc309314367)

[5.3 Kiến trúc framework 26](#_Toc309314368)

[5.3.1 Kiến trúc tổng quan framework 26](#_Toc309314369)

[5.3.2 Matching tool 26](#_Toc309314370)

[5.3.3 Data mining 26](#_Toc309314371)

[5.4 Mô hình kiến trúc Job Zoom có nhiều ưu điểm vượt trội 27](#_Toc309314372)

[5.4.1 Kiến trúc linh hoạt và khả năng mở rộng của framework 27](#_Toc309314373)

[5.4.2 Khả năng ứng dụng vào những ngành nghề khác nhau 27](#_Toc309314374)

[5.4.3 Khả năng ứng dụng vào những lĩnh vực khác 27](#_Toc309314375)

[5.4.4 Triển khai nhanh chóng 27](#_Toc309314376)

[5.4.5 Đảm bảo hiệu năng hệ thống 27](#_Toc309314377)

[6 Giải pháp công nghệ 27](#_Toc309314378)

[6.1 .NET Framework 27](#_Toc309314379)

[6.2 SQL Server Code Name “Denali” CTP3 28](#_Toc309314380)

[6.2.1 Analysis Service 28](#_Toc309314381)

[6.2.2 Database Engine 28](#_Toc309314382)

[6.2.3 MDX/DMX 28](#_Toc309314383)

[6.2.4 Pivot table 28](#_Toc309314384)

[6.3 jQuery 28](#_Toc309314385)

[6.4 MVC 28](#_Toc309314386)

[7 So sánh Job Zoom với các website tìm việc hiện tại 28](#_Toc309314387)

[8 Đánh giá và hướng phát triển 28](#_Toc309314388)

[8.1 Những điểm làm được 28](#_Toc309314389)

[8.2 Những điểm hạn chế 28](#_Toc309314390)

[8.3 Hướng phát triển 28](#_Toc309314391)

[8.3.1 Xác thực độ tin cậy của CV 28](#_Toc309314392)

[8.3.2 Sematic web 29](#_Toc309314393)

[8.3.3 Phân cụm dữ liệu 29](#_Toc309314394)

[8.3.4 Phỏng vấn trực tuyến 29](#_Toc309314395)

[8.3.5 Đưa thông tin khách quan để training cho chương trình 29](#_Toc309314396)

[8.3.6 Thu thập thông tin việc làm tự động 29](#_Toc309314397)

[TÀI LIỆU THAM KHẢO 30](#_Toc309314398)

**DANH MỤC HÌNH ẢNH**

[Hình 1. Ví dụ về cây quyết định 9](#_Toc309314399)

[Hình 2. Phân chia bảng D thành những tập con Si 14](#_Toc309314400)

[Hình 3. Phân chia Si dựa vào phân lớp 15](#_Toc309314401)

[Hình 4. Độ lợi thông tin thuộc tính Travel cost/Km 16](#_Toc309314402)

[Hình 5. Độ lợi thông tin các thuộc tính còn lại trong Bảng 1 18](#_Toc309314403)

[Hình 6. Node gốc của cây quyết định sau lần lặp đầu tiên 19](#_Toc309314404)

[Hình 7. Bảng D được phân chia sau lần lặp đầu tiên 19](#_Toc309314405)

[Hình 8. Cây quyết định sau lần lặp đầu tiên 20](#_Toc309314406)

[Hình 9. Dữ liệu cho lần phân lớp thứ 2 20](#_Toc309314407)

[Hình 10. Tính Impurity degree cho lần phân lớp thứ 2 21](#_Toc309314408)

[Hình 11. Tính Impurity degree các thuộc tính cho lần phân lớp thứ 2 21](#_Toc309314409)

[Hình 12. Bảng dữ liệu sau khi chia theo phân lớp Gender 22](#_Toc309314410)

[Hình 13. Cây quyết định sau lần phân lớp thứ 2 22](#_Toc309314411)

[Hình 14. Cây quyết định đầy đủ sau 3 lần phân lớp 23](#_Toc309314412)

**DANH MỤC BẢNG**

[Bảng 1. Ví dụ bảng dữ liệu lựa chọn phương tiện di chuyển 8](#_Toc309314413)

[Bảng 2. Ví dụ về bảng dữ liệu cần dự đoán phương tiện di chuyển 1 9](#_Toc309314414)

[Bảng 3. Ví dụ về bảng dữ liệu cần dự đoán phương tiện di chuyển 2 10](#_Toc309314415)

[Bảng 4. Lợi ích khi chia bảng D theo thuộc tính “Travel cost/km” 17](#_Toc309314416)

[Bảng 5. Kết quả độ lợi thông tin sau khi phân chia bảng D theo từng thuộc tính 19](#_Toc309314417)

[Bảng 6. Bảng dữ liệu cho lần phân lớp thứ 3 23](#_Toc309314418)

**DANH MỤC BIỂU ĐỒ**

[Biểu đồ 1. Mối liên hệ giữa giá trị Entropy cực đại và số lượng phân lớp 12](#_Toc309314419)

[Biểu đồ 2. Mối liên hệ giữa giá trị Gini index cực đại và số lượng phân lớp 13](#_Toc309314420)

**THUẬT NGỮ**

# Nhập đề

Ngày nay, internet đã dần chiếm hữu cuộc sống hiện tại, hết thảy mọi việc đều được “số hoá” cũng là lúc tuyển dụng và người tìm việc đều chọn các website tìm việc làm cầu nối cho mình.

Theo số liệu thống kê của tổ chức Internet Usage World Stats[[1]](#footnote-1), tính đến nay có hơn 2,1 tỉ người trên thế giới sử dụng Internet. Bên cạnh đó, nghiên cứu về người tiêu dùng do Mintel International Group Ltd. (tổ chức quốc tế chuyên nghiên cứu về thị trường và người tiêu dùng) tiến hành cho thấy có đến 30% số người dùng Internet đã tham khảo các website tuyển dụng trực tuyến để tìm kiếm thông tin về việc làm. Chúng ta có thể thấy xu hướng tìm việc trực tuyến đang ngày càng phát triển với tốc độc rất nhanh.

Việt Nam là quốc gia đang phát triển với dân số trên 90 triệu người với trên 28,6 triệu người sử dụng internet chiếm khoảng 31,6% dân số (2011)[[2]](#footnote-2). Các trang web về lao động việc làm ra đời đã ngày càng đem lại nhiều thông tin tổng hợp, đa dạng về công việc, mở ra những cơ hội lớn hơn cho người tìm việc, đồng thời giúp doanh nghiệp giải quyết nhu cầu nhân sự nhanh chóng và hiệu quả hơn. Tuy nhiên, thực trạng chung của các website tìm việc hiện nay chưa ứng dụng triệt để khai thác dữ liệu trong việc việc gợi ý người dùng viết CV hay hỗ trợ doanh nghiệp trong việc đánh giá ứng viên thông qua thông tin ứng viên cung cấp. Để góp phần đáp ứng những yêu cầu thực tiễn này, đề tài “**xây dựng kiến trúc hệ thống cổng thông tin tìm việc**” được ra đời.

Phương pháp thực hiện của chúng tôi là phân tích hệ thống các website tìm việc trong và ngoài nước nhóm, cùng với việc thu thập sưu liệu các thông tin hiện có trên internet, các mẫu thiết kế và kiến trúc cơ bản, lý thuyết khai thác dữ liệu và phương pháp phân loại động để đánh giá điểm mạnh, điểm yếu của hệ thống các website tìm việc hiện có, từ đó đưa ra những đưa ra những công cụ hỗ trợ người dùng và phương pháp so sánh đánh giá ứng viên.

Mục tiêu của nhóm chúng tôi khi thực hiện đề tài “**xây dựng kiến trúc hệ thống cổng thông tin tìm việc”** là cung cấp một framework đảm bảo tính linh hoạt khi áp dụng trong nhiều hệ thống ngành nghề khác nhau, hỗ trợ người tìm việc hoàn thiện CV tìm việc của mình và hỗ trợ nhà tuyển dụng trong việc đánh giá ứng viên. Đồng thời, doanh nghiệp cung cấp website tìm việc có thể dễ dàng xây dựng hoặc hoàn thiện website của mình dựa trên framework này (Job Zoom framework).

Với những ưu điểm của Job Zoom framework, chúng tôi hy vọng đề tài “**xây dựng kiến trúc hệ thống cổng thông tin tìm việc”** sẽ được ứng dụng rộng rãi trong các website tìm việc trực tuyến, góp phần nâng cao hiệu quả và tiện ích cho người tìm việc và doanh nghiệp tuyển dụng.

# 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 tìm việc đầy đủ và rõ ràng đối với ngành nghề mà người tìm 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 tìm 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 chi phí.

Với Job Zoom framework, 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

* Xây dựng kiến trúc phần mềm đảm bảo tính linh hoạt khi áp dụng trong nhiều hệ thống ngành nghề khác nhau.
* Kết hợp được với các phương pháp khai thác dữ liệu cơ bản tạo nền tảng cải thiện khả năng thích ứng của hệ thống với từng loại ngành nghề.
* Xây dựng được kiến trúc phần mềm đảm bảo hiệu năng của hệ thống với số lượng người sử dụng lớn

# Giải quyết bài toán

## Khái quát phương pháp giải quyết bài toán

Sau thời gian nghiên cứu và đánh giá một số phương pháp khai thác dữ liệu, chúng tôi thống nhất sử dụng ***cây quyết định*** trong việc giải quyết bài toán 1, 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, nhằm gợi ý người dùng viết một CV đơn giản, phù hợp với công việc hiện tại mà người tìm 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 tìm 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

Đối với bài toán 2 và 3, 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 tìm 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.

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… Để giải quyết bài toán này, Job Zoom framework sẽ tạo view và khai thác dữ liệu tự động trên cụm máy chủ khác, sau khi có kết quả, Job Zoom framework sẽ lưu dữ liệu đã khai thác vào cơ sở dữ liệu chính của chương trình. Quá trình này được thực hiện hoàn toàn độc lập sau một khoảng thời gian hay dữ liệu phát triển lên đến một độ lớn nhất định.

## 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 thường có kiểu dữ liệu là binary, nominal, ordinal, continuos.
* Ví dụ “Phương tiện di chuyển”: 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 8), 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 đào tạo (training data)
* Ví dụ: dựa vào ví dụ ở phần 3.1.1 Giới thiệu về cây quyết định (trang 8). 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 9), 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):**
  + Có sự khác nhau giữa các cách tính mức độ pha trộn “Impurity degree” giữa bảng D và các tập con Si, 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 so sánh lợi ích khi chia bảng dữ liệu theo các giá trị của thuộc tính, từ đó chọn ra thuộc tính tối ưu để phân chia
* 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. Bảng trên cho thấy, độ lợi thông tin khi phân lớp theo thuộc tính “Travel cost/km” là cao nhất nên chúng ta sẽ chọn “Travel cost/km” là node gốc của cây.



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 giải thuật xây dựng cây quyết định

#### Giải thuật ID3

ID3 là một giải thuật xây dựng cây quyết định đơn giản nhưng nó chỉ phù hợp với một lớp các bài toán hay vấn đề có thể biểu diễn bằng ký hiệu. Giải thuật này thuộc tiếp cận giải quyết vấn đề dựa trên ký hiệu (symbol – based approach).

Tập dữ liệu rèn luyện ở đây bao gồm các ví dụ được mô tả bằng các cặp “Thuộc tính – giá trị”, như trong ví dụ “Phương tiện di chuyển” trình bày trong suốt phần 3.2, đó là “Income level – Low”, “Income level – Medium” hay “Income level – High”… và mỗi ví dụ đều có một thuộc tính phân loại, ví dụ như ‘Transportation mode”, thuộc tính này phải có giá trị rời rạc, như Bus, Train hay Car

Tuy nhiên, khác với một số giải thuật khác cũng thuộc tiếp cận này, ID3 sử dụng các ví dụ rèn luyện ở dạng xác suất nên nó có ưu điểm là ít bị ảnh hưởng bởi một vài dữ liệu nhiễu. Vì vậy, tập dữ liệu rèn luyện ở đây có thể chứa lỗi hoặc có thể thiếu một vài giá trị ở một số thuộc tính nào đó. Một giải pháp thường được áp dụng đối với các dữ liệu bị thiếu là sử dụng luật đa số, chương trình tiền xử lý dữ liệu sẽ điền vào các vị trí còn trống giá trị có tần số xuất hiện cao nhất của thuộc tính đó.

ID3 còn được thảo luận nhiều vấn đề liên quan như làm sao để tránh cho cây quyết định không bị ảnh hưởng quá nhiều (overfitting) vào dữ liệu rèn luyện, để nó có thể tổng quát hơn, phân loại đúng được cho các trường hợp chưa gặp. Có nhiều giải pháp đã được đưa ra như cắt tỉa lại cây quyết định sau khi học, hoặc cắt tỉa các luật sau khi chuyển cây về dạng luật. Một vấn đề khác nữa đó là nếu như một vài thuộc tính nào đó có giá trị liên tục thì sao. Giải quyết các vấn đề này dẫn đến việc sinh ra nhiều thế hệ sau của ID3, một giải thuật nổi bật trong số đó là C4.5 (Quinlan 1996). Ngoài ra, một số kỹ thuật được tạo ra để thao tác trên dữ liệu nhằm tạo ra các cây quyết định khác nhau trên cùng tập dữ liệu rèn luyện đã cho như kỹ thuật bagging and boosting.

#### Giải thuật C4.5

C4.5 là sự mở rộng của giải thuật ID3

### 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 mục 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 có 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

Đối với bài toán của chúng tôi, thuộc tính phân lớp “Is Approve” (thuộc tính xác định ứng viên có được nhà tuyển dụng nhận hay không sau khi apply) đóng vai trò là thuộc tính phân lớp.

Như vậy, khi khai thác dữ liệu, hệ thống sẽ sinh ra rất nhiều cây quyết định theo từng vị trí làm việc, những cây này sẽ được gom lại giúp việc truy vấn kết quả trở nên dễ dàng hơn. Node gốc của cây là lĩnh vực ngành nghề, đối với từng lĩnh vực ngành nghề sẽ có nhiều vị trí công việc khác nhau.

Kết quả khi tạo cây quyết định thể hiện chúng ta thấy mức độ phụ thuộc, tầm quan trọng của các thuộc tính trong việc lựa chọn ứng viên vào một ví trí cụ thể. Những thuộc tính này gợi ý cho người dùng những kỹ năng, kiến thức người tìm việc nên có đối với vị trí làm việc mà họ đang tìm. Như vậy, khi người dùng sử dụng chương trình, chương trình sẽ đặt câu hỏi và liệt kê các thuộc tính (kỹ năng, kinh nghiệm) có mức độ phụ thuộc cao vào xác suất ứng viên được nhận, ứng viên có thể thay đổi CV cho phù hợp với vị trí mình đang cần ứng tuyển. Nếu một trong những thuộc tính mà Job Zoom gợi ý người tìm việc không thể đáp ứng được, thì Job Zoom có thể gợi ý cho ứng viên những thuộc tính khác hoặc ứng viên có thể học tập, rèn luyện để bổ sung kịp thời những thuộc tính đó nhằm tăng xác suất và tính cạnh tranh của họ khi apply vào công việc.

Thông qua cây quyết định và việc gom nhóm cây, bài toán 1 của chúng tôi đã được giải quyết. Tuy nhiên, mức độ giải quyết này đáp ứng cho những ứng viên muốn ứng tuyển vào một vị trí công việc bất kỳ, xác suất của ứng viên được nhận cao khi ứng viên apply vào nhiều công ty tuyển dụng tương ứng với vị trí đó. Việc ứng dụng taxonomy được đề cập sau đây sẽ giải quyết phần còn lại của bài toán 1, khi ứng viên apply vào vị trí công việc tại một công ty cụ thể, thì những thuộc tính nào giúp ứng viên apply vào vị trí đó có xác suất được nhận cao nhất.

### Phương pháp tiền xử lý dữ liệu khi ứng dụng cây quyết định

Để việc khai thác dữ liệu bằng cây quyết định thực hiện dễ dàng hơn, chúng tôi tiến hành xử lý dữ liệu được lưu trữ theo dạng Tagging (Taxonomy) bằng kỹ thuật Pivot Transformation trong SQL Server.

Ví dụ dữ liệu của thông tin tuyển dụng gồm vị trí đăng tuyển (Job Title), tên công ty (Company), và một tập hợp các yêu cầu/ tiêu chí, với mỗi hàng là một tiêu chí cụ thể.

|  |  |  |
| --- | --- | --- |
| **Usage** | **SetKey** | **PivotKey** |
| **Column Name** | **Job Title** | **Tag** |
| **Data**  **Records** | Developer | OOP |
| Developer | Software design |
| Developer | DBMS |
| Developer | .Net Framework |
| Tester | Testing Technique |
| Tester | .Net Framework |
| Tester | Automation Testing |
| Tester | DBMS |

Bảng 7. Dữ liệu của các thông tin tuyển dụng trước khi thực hiện Pivot trên cột Tag

Bảng sau thể hiện kết quả sau khi Pivot Transformation

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Job Title** | **OOP** | **Software design** | **DBMS** | **.Net Framework** | **Testing Technique** | **Automation Testing** |
| Data  Records | Developer | True | True | True | True | False | False |
| Tester | False | False | True | True | True | True |

Bảng 8. Kết quả sau khi Pivot Transformation

## Phân lớp phân cấp Taxonomy

### Tổng quan về taxonomy

Phân lớp là quá trình gán tự động một đối tượng vào một hoặc nhiều lớp cho trước. Trong trường hợp số lượng các lớp lớn, bài toán sẽ trở nên phức tạp hơn, chính vì vậy, khi tiến hành phân lớp thường cho kết quả có độ chính xác không cao. Một vấn đề được đặt ra là cần phân lớp những kỹ năng, kinh nghiệm của người tìm việc hay những yêu cầu công việc của nhà tuyển dụng sử dụng cấu trúc phân cấp.

Vào những năm 90 của thế kỷ XX, khái niệm taxonomy được sử dụng trong nhiều lĩnh vực khác nhau như khoa học xã hội, tâm lý học và công nghệ thông tin… để thiết lập sử trùng hợp giữa thuật ngữ của người sử dụng và thuật ngữ của hệ thống. Các chuyên gia đầu tiên phát triển cấu trúc hệ thống web đã dùng thuật ngữ taxonomy để nói đến việc tổ chức nội dung các trang web. Kể từ đó, khái niệm taxonomy được sử dụng rộng rãi với mục đích này.

Do được sử dụng trong nhiều lĩnh vực khác nhau, nên cũng có nhiều định nghĩ về taxonomy. Từ năm 2000 đến 2005 có hơn 36 định nghĩa khác nhau, dựa theo tài liệu của Hypertext[[3]](#footnote-3) chúng tôi định nghĩa taxonomy như sau:

***Taxonomy là sự phân loại toàn bộ thông tin trong một hế thống có phân cấp, sự phân loại này theo một mối quan hệ có trước của các thực thể trong thế giới thực mà nó biểu diễn.***

Một taxonomy thường được mô tả với gốc ở trên cùng, mỗi nút của taxonomy – bao gồm cả gốc – là một thực thể thông tin đại diện cho một thực thể trong thế giới thực. Giữa các nút trong taxonomy có một mối quan hệ đặc biệt gọi là ***is subclassification of*** nếu hướng liên kết từ nút con lên nút cha hoặc là ***is superclassification of*** nếu hướng liên kết từ nút cha xuống nút con. Đôi khi những quan hệ này được xác định một cách chặt chẽ hơn là ***is subclass of*** hoặc ***is superclass of***, nếu thực thể thông tin là một lớp đối tượng.

Ví dụ sau đây mô tả một taxonomy đơn giản gồm lớp Person, lớp con của nó là Employee, Manager; Lớp cha của Person là Agent. Khi đi lên từ gốc của taxonomy, các thực thể khái quát hơn. Khi đi xuống những lá ở cuối, thực thể xác định rõ ràng hơn. Ví dụ, Agent chung chung hơn Person, Employee cụ thể hơn Person.



Hình 15. Ví dụ về một taxonomy đơn giản

Taxonomy thể hiện ưu điểm trong việc phân lớp thực thể thông tin theo ngữ nghĩa, chúng thiết lập một quan hệ ngữ nghĩa đơn giản để phân biệt giữa các đối tượng trong một miền thông tin.

Taxonomy đóng vai trò rất quan trọng trong việc tổ chức thông tin và tổ chức tri thức. Nó được sử dụng chủ yếu để giúp cho việc tìm kiếm và duyệt thông tin thuận lợi và nhanh chóng hơn, đặc biệt khi chúng ta chỉ có những thông tin chung chung về vấn đề cần tìm kiếm. Khi tìm kiếm trên Internet, nếu sử dụng từ khoá để tìm kiếm thông tin, kết quả trả về có thể từ vài nghìn đến vài chục nghìn tài liệu về các chủ đề khác nhau. Sử dụng taxonomy để tìm kiếm và duyệt thông tin sẽ tiết kiệm được rất nhiều thời gian cho người dùng để tìm được thông tin cần thiết. Đồng thời, taxonomy cho phép các máy tìm kiếm và các ứng dụng có thể dễ dàng tìm được các thực thể thông tin nhanh và chính xác hơn nhiều.

### Ứng dụng taxonomy vào bài toán

Phân loại tag

## Phương pháp phân loại động

# 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

## Thuộc tính động và cây đa cấp

## Phân nhóm dữ liệu

## 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

ADO.NET Entity Framework tốc độ so với Linq

# Giải pháp công nghệ

## .NET Framework

## SQL Server Code Name “Denali” CTP3

### Database Engine

### Analysis Service

### MDX/DMX

### Pivot table

## jQuery

## MVC

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

# Đánh giá và hướng phát triển

## Những điểm làm được

## Những điểm hạn chế

Cây đa cấp

## Hướng phát triển

### Xác thực độ tin cậy của CV

### Sematic web

Tạo cây có mối quan hệ giữa các node

### Đánh trọng số cho các thuộc tính dựa vào độ sâu của taxonomy

### Phân cụm dữ liệu

### Phỏng vấn trực tuyến

Hỗ trợ người dùng thêm khâu phỏng vấn trực tuyến

### Đưa thông tin khách quan để training cho chương trình

### Thu thập thông tin việc làm tự động

**TÀI LIỆU THAM KHẢO**

[1] Lior Rokach & Oded Maimon, Data Mining with Decision Trees: Theory and Applications, 2008.

[2]Jamie MacLennan, ZhaoHui Tang & Bogdan Crivat, *Data Mining with Microsoft SQL Server 2008*, Wiley Publishing Inc., 2008

[3] Lan H. Witten, Eibe Frank and Mark A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques - Third Edition*, Elsevier, 2011

[4] Rafael Olivas, *Decision Trees - A Primer forDecision-making Professionals,* 2007

[5] Scott A. Golder & Bernardo A. Huberman, *The Structure of Collaborative Tagging Systems*

[6] Adam Freeman & Steven Sanderson, *Pro ASP.NET MVC 3 Framework – Third Edition*, Apress

[7] Tomislav Piasevoli, *MDX with Microsoft SQL Server 2008 R2 Analysis Services: Cookbook,* Packt Publishing, 2011

[8] Arttennick, *Practical MDX Queries for Microsoft SQL Server Analysis Serivces 2008*, McGraw-Hill Companies

[9] http://bis.net.vn/forums/p/378/661.aspx

[10] Nguyễn Thị Thùy Linh, *Thuật toán phân lớp cây quyết định*, Khóa luận tốt nghiệp đại học, Trường Đại học Công nghệ, 2005.

[11]

[12]

[13]

[14]

[15]

[16]

[17]

[18]

[19]

[20]

[21]

[22]

[23]

[24]

[25]

[26]

[27]

[28]

[29]

[30]

1. http://www.internetworldstats.com/stats.htm [↑](#footnote-ref-1)
2. http://www.internetworldstats.com/asia.htm#vn [↑](#footnote-ref-2)
3. http://www.hipertext.net/english/pag1011.htm#origenNota2 [↑](#footnote-ref-3)