TRƯỜNG ĐẠI HỌC THỦY LỢI

**KHOA CÔNG NGHỆ THÔNG TIN**

A blue and white logo

Description automatically generated with low confidence

**BÀI TẬP LỚN**

HỌC PHẦN: NHẬN DẠNG MẪU

**ĐỀ TÀI: DỰ ĐOÁN QUYẾT ĐỊNH NGHỈ VIỆC CỦA NHÂN VIÊN SỬ DỤNG THUẬT TOÁN CART VÀ NAIVE BAYES KẾT HỢP VỚI PCA**

Giảng viên hướng dẫn: PGS.TS. Nguyễn Quang Hoan

Sinh viên/nhóm sinh viên thực hiện:

1. Cao Thị Mai Phương 62K-ĐT3

2. Hoàng Hiểu Nhi 2154020987 - 63TTNT

3. Nguyễn Văn Luyện 2151264669 - 63TTNT

4. Nguyễn Đình Cường 2151264648 - 63TTNT

**Hà Nội, năm 2024**

|  |
| --- |
| **BỘ GIÁO DỤC VÀ ĐÀO TẠO**  **BỘ NÔNG NGHIỆP VÀ PHÁT TRIỂN NÔNG THÔN**  **TRƯỜNG ĐẠI HỌC THỦY LỢI**  CAO THỊ MAI PHƯƠNG  HOÀNG HIỂU NHI  NGUYỄN VĂN LUYỆN  NGUYỄN ĐÌNH CƯỜNG  **DỰ ĐOÁN QUYẾT ĐỊNH NGHỈ VIỆC CỦA NHÂN VIÊN SỬ DỤNG THUẬT TOÁN CART VÀ NAIVE BAYES KẾT HỢP VỚI PCA**  CHUYÊN NGÀNH: TRÍ TUỆ NHÂN TẠO  GIẢNG VIÊN HƯỚNG DẪN:  **NGUYỄN QUANG HOAN**  Hà Nội, ngày tháng 11 năm 2024 |

# **LỜI CAM ĐOAN**

Em xin cam đoan đây là bài nghiên cứu của bản thân sinh viên. Các kết quả nghiên cứu và các kết luận trong báo cáo là trung thực, không sao chép từ bất kỳ một nguồn nào và dưới bất kỳ hình thức nào. Việc tham khảo các nguồn tài liệu (nếu có) đã được thực hiện trích dẫn và ghi nguồn tài liệu tham khảo theo đúng quy định.

**Nhóm Sinh viên**

**Báo cáo bài tập lớn**

Cao Thị Mai Phương

Hoàng Hiểu Nhi

Nguyễn Văn Luyện

Nguyễn Đình Cường

# **LỜI CẢM ƠN**

Em xin gửi lòng biết ơn sâu sắc đến những tâm huyết và tri thức mà thầy cô trong Công nghệ thông tin– Trường Đại học Thủy Lợi đã chỉ bảo cho nhóm em. Đặc biệt em xin gửi lời cảm ơn đến thầy Nguyễn Quang Hoan – Người đã tận tình hướng dẫn, hỗ trợ về chuyên môn và theo dõi sát sao quá trình làm bài tập lớn của nhóm em.

Trong quá trình học tập nhóm em đã học hỏi được nhiều kinh nghiệm hơn trong công việc nhưng vẫn không thể tránh khỏi các sai sót.Nhóm em rất mong sẽ nhận được các ý kiến đóng góp quý báu của thầy/cô cũng như ban lãnh đạo của công ty để nhóm em có thể hoàn thiện hơn trong tương lai.

Sau cùng, nhóm em xin kính chúc thầy cô trong Khoa Công nghệ thông tin dồi dào sức khỏe và công tác tốt.

MỤC LỤC

[LỜI CAM ĐOAN 3](#_Toc181606532)

[LỜI CẢM ƠN 4](#_Toc181606533)

[MỤC LỤC 5](#_Toc181606534)

[CHƯƠNG I: GIỚI THIỆU TỔNG QUAN BÀI TOÁN 6](#_Toc181606535)

[CHƯƠNG 2: CÁC KĨ THUẬT SỬ DỤNG TRONG BÀI TOÁN 7](#_Toc181606536)

[2.1. Tổng quan về học máy 7](#_Toc181606537)

[**2.1.1. Khái niệm học máy** 7](#_Toc181606538)

[**2.1.2. Phân loại nhóm thuật toán trong học máy** 8](#_Toc181606539)

[**2.1.3. Các mô hình sử dụng trong bài toán** 8](#_Toc181606540)

[Ưu điểm 14](#_Toc181606541)

[2.2. Các phương pháp đánh giá mô hình 15](#_Toc181606542)

[CHƯƠNG 3: THỰC NGHIỆM 17](#_Toc181606543)

[3.1. Mô tả bài toán 17](#_Toc181606544)

[3.2. Tiền xử lý dữ liệu 18](#_Toc181606545)

[***3.2.1.*** ***Mã hoá dữ liệu*** 18](#_Toc181606546)

[***3.2.3.*** ***Xử lý mất cân bằng dữ liệu*** 19](#_Toc181606547)

[3.3. Huấn luyện mô hình 20](#_Toc181606548)

[***3.3.1. Hàm đánh giá tỉ lệ dự đoán đúng*** 20](#_Toc181606549)

[***3.3.2.***  ***Hàm show tỉ lệ của thuật toán CART và Naive Bayes*** 21](#_Toc181606550)

[***3.3.4. Mô hình Naive CART PCA*** 21](#_Toc181606551)

[***3.3.3.***  ***Mô hình Naive Bayes\_PCA*** 22](#_Toc181606552)

[3.4. Đánh giá 22](#_Toc181606553)

[3.5. Giao diện 23](#_Toc181606554)

[KẾT LUẬN 24](#_Toc181606555)

[TÀI LIỆU THAM KHẢO 25](#_Toc181606556)

# **CHƯƠNG I: GIỚI THIỆU TỔNG QUAN BÀI TOÁN**

Trong bối cảnh cạnh tranh ngày càng gia tăng trên thị trường lao động hiện đại, việc giữ chân nhân viên trở thành một thách thức then chốt đối với mọi tổ chức. Quyết định nghỉ việc của nhân viên không chỉ ảnh hưởng đến tính ổn định và hiệu suất làm việc của công ty mà còn tác động sâu sắc đến văn hóa doanh nghiệp và lòng trung thành của các nhân viên còn lại. Chính vì vậy, việc dự đoán hành vi nghỉ việc đã trở thành một nhiệm vụ quan trọng và cần thiết cho các nhà quản lý nhân sự.

Bài toán này sẽ xem xét một loạt các yếu tố có thể ảnh hưởng đến quyết định nghỉ việc, bao gồm môi trường làm việc, kinh nghiệm, đam mê nghề nghiệp, tốc độ thăng tiến, và mức lương. Để giải quyết vấn đề này, chúng tôi sẽ áp dụng các thuật toán học máy hiện đại, cụ thể là CART (Classification and Regression Trees) và Naive Bayes, kết hợp với phân tích thành phần chính (PCA). PCA sẽ giúp rút gọn dữ liệu bằng cách xác định các thành phần chính có ảnh hưởng lớn nhất đến quyết định nghỉ việc, từ đó giảm thiểu số biến cần thiết mà vẫn đảm bảo giữ lại thông tin quan trọng.

Thuật toán CART sẽ cho phép chúng tôi tạo ra các cây quyết định trực quan, giúp phân tích rõ ràng mối quan hệ giữa các yếu tố và khả năng nghỉ việc của nhân viên. Trong khi đó, Naive Bayes, với khả năng dự đoán dựa trên xác suất, sẽ mang lại một cái nhìn khác về cách các yếu tố này tương tác với nhau.

Thông qua việc áp dụng các kỹ thuật này, chúng em không chỉ muốn phát triển một mô hình dự đoán chính xác mà còn cung cấp các thông tin giá trị cho các nhà quản lý. Điều này sẽ giúp họ hiểu rõ hơn về các động lực nghỉ việc của nhân viên và xây dựng các chiến lược hiệu quả nhằm giữ chân nhân tài, từ đó cải thiện môi trường làm việc và tăng cường sự gắn bó của nhân viên đối với tổ chức.

# **CHƯƠNG 2: CÁC KĨ THUẬT SỬ DỤNG TRONG BÀI TOÁN**

## **2.1. Tổng quan về học máy**

### **2.1.1. Khái niệm học máy**

Học máy (Machine Learning) là một lĩnh vực của trí tuệ nhân tạo (AI) tập trung vào việc phát triển các thuật toán cho phép máy tính học hỏi từ dữ liệu. Thay vì được lập trình cụ thể cho một nhiệm vụ, các hệ thống học máy tự cải thiện hiệu suất khi tiếp xúc với nhiều dữ liệu hơn. Học máy có thể được chia thành ba loại chính:

* Học có giám sát: Trong đó mô hình được huấn luyện trên tập dữ liệu có nhãn.
* Học không giám sát: Mô hình tìm kiếm cấu trúc và mẫu từ dữ liệu không có nhãn.
* Học bán giám sát: Kết hợp giữa học có giám sát và không giám sát, sử dụng một lượng nhỏ dữ liệu có nhãn và nhiều dữ liệu không có nhãn.

**Ưu điểm và hạn chế của Học có giám sát và không giám sát:**

* Ưu điểm:
  + Học có giám sát: Supervised learning là thuật toán dự đoán đầu ra (outcome) của một dữ liệu mới (new input) dựa trên các cặp (*input, outcome*) đã biết từ trước. Cặp dữ liệu này còn được gọi là (*data, label*), tức (*dữ liệu, nhãn*). Supervised learning là nhóm phổ biến nhất trong các thuật toán Machine Learning. Chúng học thông qua một bộ dữ liệu đã được gán nhãn chính xác và cố gắng đưa ra các dự đoán đầu ra mục tiêu chính xác nhất có thể trên bộ dữ liệu cho trước.
  + Học không giám sát: Thuật toán unsupervised learning sẽ dựa vào cấu trúc của dữ liệu để thực hiện một công việc nào đó, ví dụ như phân nhóm (clustering) hoặc giảm số chiều của dữ liệu (dimension reduction) để thuận tiện trong việc lưu trữ và tính toán.
* Hạn chế:
  + Có một phương pháp học rơi vào giữa 2 phương pháp này là học bán giám sát (semi-supervised learning) khi dữ liệu học chỉ có một phần nhỏ là có đầy đủ cặp đầu vào - đầu ra tương ứng và phần dữ liệu còn lại chỉ có đầu vào. Phương pháp này phù hợp trong trường hợp việc gắn đầu ra cho từng đầu vào là không khả thi hoặc tốn rất nhiều tài nguyên và việc sử dụng dữ liệu không được gắn đầu ra sẽ hiệu quả hơn.
  + Học không giám sát: chúng ta không biết câu trả lời chính xác cho mỗi dữ liệu đầu vào. Giống như khi ta học, không có thầy cô giáo nào chỉ cho ta biết đó là chữ A hay chữ B. Cụm *không giám sát* được đặt tên theo nghĩa này. Các dữ liệu không được "hướng dẫn" trước như trong trường hợp học có giám sát. Các thuật toán cần học được từ dữ liệu mà không hề có bất cứ sự hướng dẫn nào.

### **2.1.2. Phân loại nhóm thuật toán trong học máy**

Các thuật toán trong học máy có thể được phân loại thành nhiều nhóm, bao gồm:

* Thuật toán hồi quy: Dùng để dự đoán giá trị liên tục. Ví dụ: hồi quy tuyến tính.
* Thuật toán phân loại: Dùng để phân loại dữ liệu thành các nhóm. Ví dụ: Naive Bayes, CART (Classification and Regression Trees).
* Thuật toán phân cụm: Nhằm phân nhóm dữ liệu mà không cần nhãn. Ví dụ: K-means.
* Thuật toán giảm chiều: Giúp giảm số lượng biến trong dữ liệu, giữ lại các thông tin quan trọng. Ví dụ: PCA (Principal Component Analysis).

### **2.1.3. Các mô hình sử dụng trong bài toán**

Trong bài toán dự đoán quyết định nghỉ việc của nhân viên, chúng tôi sẽ sử dụng các mô hình sau:

***CART (Classification and Regression Trees):***

Cart là một thuật toán phân loại mạnh mẽ, CART xây dựng cây quyết định để phân loại các đối tượng dựa trên các thuộc tính đầu vào. Cây quyết định giúp dễ dàng hiểu và giải thích các quyết định mà mô hình đưa ra.

***Naive Bayes:***

Là một thuật toán phân loại dựa trên định lý Bayes, giả định rằng các thuộc tính đầu vào độc lập với nhau. Thuật toán này rất hiệu quả trong các bài toán phân loại văn bản và có thể hoạt động tốt ngay cả khi dữ liệu có nhiều thuộc tính.

***PCA (Principal Component Analysis):***

Được sử dụng để giảm chiều dữ liệu, PCA giúp loại bỏ các thuộc tính không cần thiết và giữ lại các thông tin quan trọng nhất. Điều này không chỉ giúp tăng tốc độ xử lý mà còn cải thiện độ chính xác của các mô hình học máy bằng cách giảm thiểu hiện tượng overfitting.

#### *2.1.3.1. Thuật toán CART*

Thuật toán CART (cây phân loại và hồi quy - Classification and Regression Tree), là một thuật toán được Leo Breiman giới thiệu năm 1984, dùng để xây dựng cây quyết định đa năng, với mục đích phân loại và hồi quy dữ liệu.

Thuật toán CART là nền tảng cho những thuật toán khác mạnh mẽ hơn như bagged decision trees (tập hợp nhiều cây quyết định để cùng đoán nhận), random forest (rừng ngẫu nhiên) và boosted decision trees (tạp hợp các cây khác nhau, là các model; và một siêu model - meta model - để tập hợp kết quả đoán nhận của các model con)

Chúng ta có thể hiểu cây quyết định một cách đơn giản là một đồ thị có hình dạng giống như cây và nó có hướng được sử dụng để cho việc ra quyết định. Ví dụ bạn cần đưa ra quyết định của mình có nên đi chơi hay không khi phụ thuộc vào các yếu tố như thời tiết, nhiệt độ, đổ ẩm, gió…

Biểu diễn của mô hình CART

Biểu diễn của mô hình CART chính là một cây nhị phân.

Cây nhị phân này cũng chính là cây nhị phân thường thấy trong môn cấu trúc dữ liệu và giải thuật, với mỗi nút là một biến độc lập riêng lẻ

x (trong học máy, ta gọi đây là các đặc trưng), và điểm tách (hay là ngưỡng phân loại) của đặc trưng đó.

Nút lá của cây chứa biến đầu ra y, dùng để thực hiện đoán nhận.

Thuật toán

Trong cây quyết định, các nút được chia thành các nút con dựa trên giá trị ngưỡng của một thuộc tính. Thuật toán CART thực hiện điều đó bằng cách tìm kiếm tính đồng nhất tốt nhất cho các nút con, với sự trợ giúp của tiêu chí Chỉ số Gini.

Nút gốc được lấy làm tập huấn luyện và được chia thành hai bằng cách xem xét giá trị ngưỡng và thuộc tính tốt nhất. Hơn nữa, các tập con cũng được chia bằng cùng một logic. Điều này tiếp tục cho đến khi tập con thuần túy cuối cùng được tìm thấy trong cây hoặc số lượng lá tối đa có thể có trong cây đang phát triển đó. Điều này cũng được gọi là Cắt tỉa cây.

Các mô hình CART từ Data:

Theo [Machine Learning Mastery](https://machinelearningmastery.com/classification-and-regression-trees-for-machine-learning/) , mô hình CART được hình thành bằng cách chọn các biến đầu vào và đánh giá các điểm phân chia trên các biến đó cho đến khi tạo ra được một cây thích hợp. Chúng ta hãy xem xét các bước cần thiết để tạo Cây quyết định bằng thuật toán CART:

● Thuật toán tham lam :

Các biến đầu vào và các điểm chia tách được chọn thông qua thuật toán tham lam. Xây dựng cây quyết định nhị phân là một kỹ thuật chia tách không gian đầu vào. Một điều kiện kết thúc được xác định trước, chẳng hạn như số lượng ví dụ đào tạo tối thiểu được cung cấp cho mỗi nút lá của cây, được sử dụng để dừng việc xây dựng cây.

Không gian đầu vào được chia bằng cách sử dụng phương pháp Greedy. Phương pháp này được gọi là phân tách nhị phân đệ quy. Đây là phương pháp số trong đó tất cả các giá trị được căn chỉnh và một số điểm phân tách được thử và đánh giá bằng cách sử dụng hàm chi phí, với phép phân tách có chi phí thấp nhất được chọn.

Hàm [chi phí](https://www.analyticssteps.com/blogs/cost-function-machine-learning) được giảm xuống để xác định điểm chia tách cho các vấn đề mô hình hóa dự đoán hồi quy là tổng bình phương lỗi trên tất cả các mẫu đào tạo nằm bên trong hình chữ nhật:

tổng(y – p)^2

Tại đây, y là đầu ra của mẫu đào tạo và p là đầu ra ước tính cho hình chữ nhật.

Hàm chỉ số Gini được sử dụng để phân loại và nó chỉ ra mức độ "tinh khiết" của các nút lá. Công thức cho việc này là:

G = tổng(pk \* (1 – pk))

Tại đây, G là chỉ số Gini, pk là tỷ lệ các trường hợp đào tạo có lớp k trong hình chữ nhật .

● Tiêu chí dừng:

Khi thực hiện quá trình phân tách nhị phân đệ quy theo cây dữ liệu đào tạo, phương pháp phân tách nhị phân đệ quy được mô tả ở trên phải biết khi nào thì dừng phân tách.

Phương pháp dừng thường xuyên nhất là sử dụng một lượng dữ liệu đào tạo tối thiểu được phân bổ cho mỗi nút lá. Nếu số lượng ít hơn một ngưỡng nhất định, thì việc chia tách sẽ bị từ chối và nút được coi là nút lá cuối cùng. Số lượng thành viên đào tạo được điều chỉnh theo tập dữ liệu. Nó chỉ định mức độ chính xác của cây đối với dữ liệu đào tạo.

● Cắt tỉa cây:

Độ phức tạp của cây quyết định được định nghĩa là số lượng các nhánh trong cây. Nên sử dụng cây có ít nhánh hơn. Chúng dễ nắm bắt và ít có khả năng phân cụm dữ liệu.

Làm việc qua từng nút lá trong cây và đánh giá tác động của việc xóa nó bằng cách sử dụng một tập kiểm tra hold-out là cách tiếp cận cắt tỉa nhanh nhất và đơn giản nhất. Chỉ các nút lá bị loại bỏ nếu tổng hàm chi phí cho toàn bộ tập kiểm tra giảm. Khi không thể đạt được cải tiến bổ sung nào nữa, thì không nên xóa thêm nút nào nữa.

Có thể áp dụng các phương pháp cắt tỉa tiên tiến hơn, chẳng hạn như cắt tỉa độ phức tạp về chi phí (còn gọi là cắt tỉa liên kết yếu nhất), trong đó tham số học (alpha) được sử dụng để xác định xem có thể loại bỏ các nút hay không tùy thuộc vào kích thước của cây con.

Thuật toán CART không phải là thuật toán tham số, do đó nó không phụ thuộc vào thông tin từ một loại phân phối nhất định.

Thuật toán CART kết hợp cả hai thử nghiệm với bộ dữ liệu thử nghiệm và xác thực chéo để đo lường chính xác hơn mức độ phù hợp.

CART cho phép sử dụng cùng một biến nhiều lần ở nhiều vùng khác nhau của cây. Kỹ năng này có khả năng tiết lộ mối quan hệ phụ thuộc phức tạp giữa các nhóm biến.

Các giá trị ngoại lệ trong các biến đầu vào không có tác động có ý nghĩa đến CART.

Người ta có thể nới lỏng các hạn chế dừng để cho phép cây quyết định phát triển quá mức và sau đó cắt tỉa cây xuống kích thước lý tưởng của nó. Phương pháp này làm giảm khả năng thiếu cấu trúc thiết yếu trong tập dữ liệu bằng cách kết thúc quá sớm.

Để chọn tập hợp biến đầu vào, CART có thể được sử dụng kết hợp với các thuật toán dự đoán khác.

**Các bước thực hiện của thật toán Cart**

* Input: tập dữ liệu đã được gán nhãn
* Output: cây quyết định của bài toán
* Là một kỹ thuật phi tham số hữu ích có thể được sử dụng để giải thích một biến phụ thuộc liên tục hoặc phân loại dưới dạng nhiều biến độc lập. Các biến độc lập có thể liên tục hoặc phân loại. CART sử dụng phương pháp phân vùng thường được gọi là “**chia để trị**”. Thuật toán CART hoạt động để tìm biến độc lập tạo ra nhóm đồng nhất tốt nhất khi tách dữ liệu.
* Công việc thực hiện của thuật toán:
  + Sử dụng chỉ số “gini index” làm thước đo trong khi tạo cây quyết định (decision tree) trong thuật toán CART (Classification and Regression Tree), chỉ số “gini index” càng cao càng tốt.
* Các bước thực hiện:
  + **B1:** Tính giá trị Gini

Gini =

* + - C: số lớp cần phân loại (số nhãn trong 1 thuộc tính)
    - N: là tổng số lượng phần tử ở node đó
* **B2:** Tính chỉ số gini\_index.

Gini\_index = gini(p) - gini()

* + - gini(p): chỉ số gini ở node cha
    - K: số node con được tách ra
    - gini(ck): chỉ số gini ở node con thứ k
    - M: số phần tử ở node p
    - mi : là số phần tử ở node con thứ i
* chọn ra thuộc tính có gini\_index lớn nhất 🡪 Ginisplit nhỏ nhất

gini()

* **B3:** Chia dữ liệu vào các nút con tương ứng với các giá trị của thuộc tính, thực hiện lặp tìm ra thuộc tính tốt nhất trong tập dữ liệu mới.
* Điều kiện dừng:

*TH1*: nếu tất cả các mẫu trong node con thuộc cùng 1 lớp C (entropy=0) thì node đó đc gán nhãn C

*TH2*: nếu node con là rỗng (không có dữ liệu) thì node đó sẽ đc gán = nhãn phổ biến nhất trong tập S

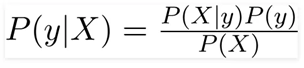
*TH3:* nếu không còn thuộc tính nào để phân chia (các thuộc tính đều đc xét hết) -> node lá đó sẽ đc gán = nhãn phổ biến nhất

#### *2.1.3.2. Thuật toán Naivie Bayes*

Naive Bayes Classification (NBC) – thuật toán phân loại Naive Bayes - là một thuật toán dựa trên định lý Bayes về lý thuyết xác suất để đưa ra các phán đoán cũng như phân loại dữ liệu dựa trên các dữ liệu được quan sát và thống kê, được ứng dụng rất nhiều trong các lĩnh vực Machine learning dùng để đưa các dự đoán có độ chính xác cao, dựa trên một tập dữ liệu đã được thu thập. NBC thuộc vào nhóm học máy có giám sát.

Mô hình phân lớp Naive Bayes

Naive Bayes là một thuật toán phân lớp được mô hình hoá dựa trên định lý Bayes trong xác suất thống kê:



Trong đó:

● P(y|X) gọi là posterior probability: xác suất của mục tiêu y với điều kiện có đặc trưng X

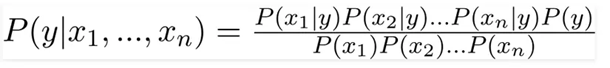
● P(X|y) gọi là likelihood: xác suất của đặc trưng X khi đã biết mục tiêu y

● P(y) gọi là prior probability của mục tiêu y

● P(X) gọi là prior probability của đặc trưng X

Ở đây, X là vector các đặc trưng, có thể viết dưới dạng:



Khi đó, đẳng thức Bayes trở thành:

Trong mô hình Naive Bayes, có hai giả thiết được đặt ra:

1. Các đặc trưng đưa vào mô hình là độc lập với nhau. Tức là sự thay đổi giá trị của một đặc trưng không ảnh hưởng đến các đặc trưng còn lại.
2. Các đặc trưng đưa vào mô hình có ảnh hưởng ngang nhau đối với đầu ra mục tiêu.

Khi đó, kết quả mục tiêu y để P(y|X) đạt cực đại trở thành:

Chính vì hai giả thiết gần như không tồn tại trong thực tế trên, mô hình này mới được gọi là naive (ngây thơ). Tuy nhiên, chính sự đơn giản của nó với việc dự đoán rất nhanh kết quả đầu ra khiến nó được sử dụng rất nhiều trong thực tế trên những bộ dữ liệu lớn, đem lại kết quả khả quan. Một vài ứng dụng của Naive Bayes có thể kể đến như: lọc thư rác, phân loại văn bản, dự đoán sắc thái văn bản, ...

Xét một bộ dữ liệu đơn giản về việc đi làm muộn của một bạn nhân viên. Bộ dữ liệu được biểu diễn dạng bảng dưới đây:

Bằng việc lập bảng tần suất (frequency table) cho từng đặc trưng theo mục tiêu, có thể tính được likelihood P(X|y). Giả sử, để dự đoán cho một ngày X=(Muộn, Xấu, Mưa), cần tính:

P(Muộn|X) ∝ P(Muộn|Muộn) \* P(Xấu|Muộn) \* P(Mưa|Muộn) \* P(Muộn) = (3/5) \* (2/5) \* (1/5) \* (5/10) = 0.024

P(Không muộn|X) ∝ P(Muộn|Không muộn) \* P(Xấu|Không muộn) \* P(Mưa|Không muộn) \* P(Không muộn) = (0/5) \* (3/5) \* (1/5) \* (5/10) = 0

=> y = argmax{ P(X|y)P(y) } = Muộn.

Vì vậy, nếu anh chàng nhân viên thấy báo thức reo rồi mà còn cố ngủ thêm 15 phút, lúc dậy thân thể uể oải, nhìn ra ngoài trời thấy đang mưa, thì khả năng rất cao là hôm nay sẽ bị trừ lương.

Ứng dụng của thuật toán Naïve Bayes

Thuật toán Naive Bayes Classification được áp dụng vào các loại ứng dụng sau

* Real time Prediction: NBC chạy khá nhanh nên nó thích hợp áp dụng ứng dụng nhiều vào các ứng dụng chạy thời gian thực, như hệ thống cảnh báo phát hiện sự cố...
* Multi class Prediction: Nhờ vào định lý Bayes mở rộng ta có thể ứng dụng vào các loại ứng dụng đa dự đoán, tức là ứng dụng có thể dự đoán nhiều giả thuyết mục tiêu.
* Text classification/ Spam Filtering/ Sentiment Analysis: NBC cũng rất thích hợp cho các hệ thống phân loại văn bản hay ngôn ngữ tự nhiên vì tính chính xác của nó lớn hơn các thuật toán khác. Ngoài ra các hệ thống chống thư rác cũng rất ưu chuộng thuật toán này. Và các hệ thống phân tích tâm lý thị trường cũng áp dụng NBC để tiến hành phân tích tâm lý người dùng ưu chuộng hay không ưu chuộng các loại sản phẩm nào từ việc phân tích các thói quen và hành động của khách hàng.
* Recommendation System: Naive Bayes Classifier được sử dụng rất nhiều để xây dựng hệ thống gợi ý.

## **Ưu điểm**

Giả định độc lập: hoạt động tốt cho nhiều bài toán/miền sữ liệu và ứng dụng.  
 Đơn giản nhưng đủ tốt để giải quyết nhiều bài toán như phân lớp văn bản, lọc spam,..

Cho phép kết hợp tri thức tiên nghiệm (priori knowledge) và dữ liệu quan sát được (observed data).  
 Tốt khi có sự chênh lệch số lượng giữa các lớp phân loại.

Huấn luyện mô hình (ước lượng tham số) dễ và nhanh.

## **2.2.** **Các phương pháp đánh giá mô hình**

Đánh giá mô hình là bước quan trọng trong quá trình xây dựng và sử dụng các thuật toán học máy. Nó giúp chúng ta hiểu rõ hiệu năng của mô hình, từ đó đưa ra quyết định tối ưu trong việc lựa chọn mô hình.

Trong bài toán ra sử dụng các độ đo và phương pháp đánh giá sau:

* Độ chính xác ( Accuracy): là chỉ số đánh giá hiệu quả của mô hình. Nó biểu thị tỷ lệ các dự đoán đúng trên tổng số dự đoán mà mô hình thực hiện.

**Accuracy = Số dự đoán đúng / Tổng số dự đoán**

* Ưu điểm: Dễ hiểu: chỉ số này rất dễ hiểu và trực quan, cho biết một cách tổng quát về hiệu năng của mô hình.
* Nhược điểm:
* Không phù hợp với các tập dữ liệu không cân bằng.
* Không phân biệt được các loại lỗi: độ chính xác không cho biết mắc phải những lỗi nào.
* Độ chính xác (Precision) là chỉ số đánh giá hiệu năng của mô hình, đặc biệt hữu ích trong các trường hợp dữ liệu không cân bằng. Nó cho biết trong số các mẫu mà mô hình dự đoán là dương và có bao nhiêu mẫu thực sự là dương.

**Precison = Số mẫu dương dự đoán đúng / Tổng số mẫu được dự đoán là dương**

* Recall (Độ nhạy): đo lường tỷ lệ các mẫu dương thực tế (positive) mà mô hình đã dự đoán đúng. Nói cách khác, nó cho biết mô hình đã tìm thấy được bao nhiêu phần trăm trong số các mẫu dương thực sự.

**Recall = TP / (TP + FN)**

* Trong đó:

TP (True Positive): Số mẫu dương được dự đoán đúng.

FN (False Negative): Số mẫu dương bị dự đoán sai thành âm.

=> Recall cao cho thấy mô hình rất giỏi trong việc tìm ra tất cả các mẫu dương.

* F1-score là trung bình điều hòa của Precision và Recall. Nó cung cấp một thước đo cân bằng về hiệu suất của mô hình.

**F1-score = 2 \* (Precision \* Recall) / (Precision + Recall)**

=> F1-score cao cho thấy mô hình có cả Precision và Recall cao, nghĩa là mô hình vừa chính xác trong việc dự đoán dương, vừa tìm ra được hầu hết các mẫu dương.

# **CHƯƠNG 3: THỰC NGHIỆM**

## **3.1.** **Mô tả bài toán**

Với yêu cầu của bài toán, một số mục tiêu được đặt ra:

* Mục tiêu 1: Thu thập dữ liệu, tiền xử lý dữ liệu
* Mục tiêu 2: Phân tích dữ liệu
* Mục tiêu 3: Dùng các phương pháp học máy để đưa ra dự đoán quyết định nghỉ việc của nhân viên
* Mục tiêu 4: Xây dựng giao diện



Dữ liệu thô gồm 5 cột thuộc tính và 1 cột nhãn.

|  |  |
| --- | --- |
| **Thuộc tính** | **Ý nghĩa** |
| Work environment | Môi trường làm việc |
| Experience | Kinh nghiệm |
| Passion | Đam mê |
| At speedvancemend | Tốc độ |
| Salary | Mức lương |
| Quit job | Bỏ việc |

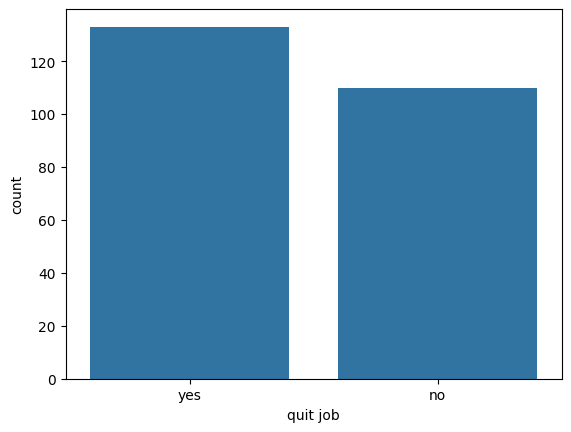
## **3.2.** **Tiền xử lý dữ liệu**

### ***3.2.1.*** ***Mã hoá dữ liệu***

Ta sử dụng Label Encoder để mã hóa các giá trị chuỗi trong các cột của dữ liệu thành các giá trị số. Sau khi mã hóa xong, ta thu được dữ liệu như hình bên dưới.

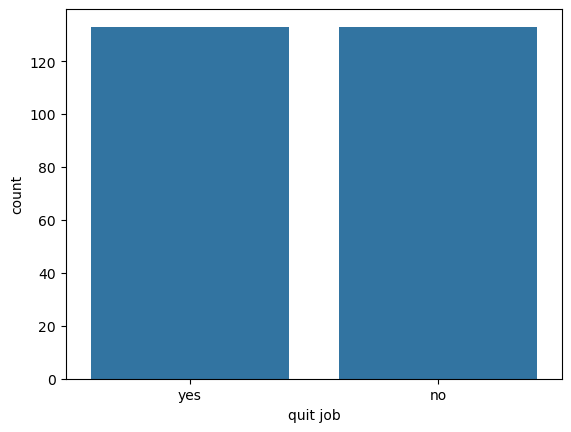


### ***3.2.3.*** ***Xử lý mất cân bằng dữ liệu***

******

Biểu đồ cho thấy một sự mất cân bằng dữ liệu giữa hai lựa chọn "yes" và "no" trong quyết định nghỉ việc. Số lượng người nghỉ việc cao hơn. Sự chênh lệch này có thể ảnh hưởng đến tính đại diện của dữ liệu nếu muốn thực hiện phân tích hoặc dự đoán dựa trên nó, vì kết quả có thể bị thiên lệch theo hướng của lựa chọn phổ biến hơn.

Sau khi thực hiện cân bằng dữ liệu bằng phương pháp SMOTE thu được biểu đồ sau:



Biểu đồ cho thấy cả hai cột "yes" và "no" hiện có số lượng mẫu gần như bằng nhau, khoảng 125 mẫu mỗi lớp, biểu thị rằng dữ liệu đã được cân bằng.Kết quả này sẽ giúp các mô hình học máy có khả năng phân loại tốt hơn giữa hai lớp, tránh việc bị thiên lệch về lớp có nhiều mẫu hơn.

## **3.3. Huấn luyện mô hình**

### ***3.3.1. Hàm đánh giá tỉ lệ dự đoán đúng***

**# Hàm tính toán tỷ lệ dự đoán chính xác**

def RateRating(Y\_Pred, Y\_test):

countPredictTrue = np.sum(Y\_Pred == Y\_test) # Đếm số lần dự đoán đúng

rate = countPredictTrue / len(Y\_Pred)

return rate

Hàm RateRating là một hàm đơn giản nhưng hiệu quả, dùng để đánh giá độ chính xác của mô hình dự đoán. Với hai mảng đầu vào là Y\_Pred (dự đoán của mô hình) và Y\_test (giá trị thực tế), hàm này tính toán tỷ lệ phần trăm dự đoán chính xác bằng cách so sánh các phần tử tương ứng trong hai mảng. Đầu tiên, hàm kiểm tra xem mỗi phần tử của Y\_Pred có bằng phần tử tương ứng của Y\_test hay không, từ đó tạo ra một mảng Boolean chứa các giá trị True và False. Sau đó, np.sum được sử dụng để đếm số lần dự đoán đúng (tức là số lượng True trong mảng Boolean). Cuối cùng, hàm chia tổng số lần dự đoán đúng cho tổng số phần tử trong Y\_Pred để tìm ra tỷ lệ dự đoán chính xác, và trả về kết quả này. Giá trị tỷ lệ chính xác là một chỉ số quan trọng để đánh giá mô hình, cho phép người dùng biết được mức độ tin cậy của các dự đoán mà mô hình đưa ra.

### ***3.3.2. Hàm show tỉ lệ của thuật toán CART và Naive Bayes***

Hàm AboutRateCART và AboutRateNB được xây dựng nhằm mục đích hiển thị hiệu suất dự đoán của thuật toán CART và Naive Bayes thông qua ba chỉ số quan trọng: Precision, Recall và F1-score. Khi gọi hàm này, một hộp thoại sẽ xuất hiện, cung cấp các thông số đánh giá chi tiết của mô hình. Đầu tiên, hàm hiển thị Accuracy score, là tỷ lệ phần trăm dự đoán chính xác của mô hình. Sau đó, hàm cung cấp Precision score, biểu thị tỷ lệ dự đoán đúng trong số các dự đoán dương. Tiếp theo là Recall score, chỉ ra tỷ lệ các trường hợp dương thật sự được dự đoán đúng. Cuối cùng là F1 score, một chỉ số tổng hợp giúp cân bằng giữa Precision và Recall. Thông qua các thông số này, người dùng có thể dễ dàng đánh giá độ chính xác và hiệu quả của thuật toán CART trong dự đoán, từ đó đưa ra những điều chỉnh phù hợp nhằm cải thiện mô hình nếu cần.

### ***3.3.4. Mô hình Naive CART PCA***

Hàm Predict With CART là một hàm dự đoán sử dụng thuật toán CART dựa trên dữ liệu đầu vào của người dùng, được thu thập từ các combobox trong giao diện. Đầu tiên, các thông tin đầu vào, bao gồm môi trường làm việc, kinh nghiệm, đam mê, tốc độ thăng tiến và mức lương, được lấy và lưu vào biến newData. Sau đó, mỗi giá trị này được mã hóa bằng LabelEncoder để chuyển từ dạng văn bản sang dạng số nhằm tương thích với mô hình. Tiếp theo, dữ liệu đã mã hóa được giảm số chiều bằng kỹ thuật PCA để đảm bảo tính hiệu quả và giảm tải tính toán cho mô hình dự đoán. Khi dữ liệu đã sẵn sàng, mô hình CART (model MaxCART) thực hiện dự đoán và kết quả được hiển thị trực tiếp trên giao diện cho người dùng. Trong trường hợp có lỗi, ví dụ như thiếu dữ liệu đầu vào, một thông báo cảnh báo sẽ được hiển thị để nhắc nhở người dùng điền đầy đủ thông tin cần thiết.

Quá trình tối ưu hóa mô hình CART bằng cách áp dụng PCA (Phân tích Thành phần Chính) để giảm kích thước dữ liệu, giúp mô hình hoạt động hiệu quả hơn. Bắt đầu với việc đặt maxRateCART = 0 để lưu trữ tỷ lệ dự đoán cao nhất, thuật toán chạy qua từng số thành phần (số chiều) khác nhau của PCA, từ 1 đến tổng số thuộc tính của dữ liệu. Trong mỗi lần lặp, thuật toán khai báo một đối tượng PCA với số thành phần cụ thể, sau đó giảm kích thước dữ liệu X\_os để giữ lại các thành phần quan trọng nhất.

Sau khi dữ liệu đã giảm kích thước, nó được chia thành tập huấn luyện và kiểm tra. Mô hình CART với tiêu chí gini (dùng để phân chia nút trong cây quyết định) được huấn luyện trên tập huấn luyện và thực hiện dự đoán trên tập kiểm tra. Nếu tỷ lệ chính xác của dự đoán cao hơn maxRateCART hiện tại, thuật toán cập nhật maxRateCART với tỷ lệ mới, đồng thời lưu lại số thành phần tối ưu (numComponentsCART), PCA tốt nhất (bestPcaCART), mô hình CART tốt nhất (modelMaxCART), và kết quả dự đoán tốt nhất (bestPredCART). Kết quả là thuật toán tìm được sự kết hợp tối ưu giữa số chiều của PCA và mô hình CART, đạt tỷ lệ dự đoán cao nhất trên tập kiểm tra.

### ***3.3.3. Mô hình Naive Bayes\_PCA***

Hàm `PredictWithNB` được thiết kế để thực hiện dự đoán bằng mô hình Naive Bayes dựa trên dữ liệu đầu vào từ người dùng. Trước tiên, các giá trị nhập vào từ giao diện như môi trường làm việc, kinh nghiệm, đam mê, tốc độ thăng tiến, và mức lương được thu thập từ các `combobox` và lưu lại trong biến `newData`. Sau đó, mỗi giá trị này được mã hóa từ dạng văn bản thành dạng số thông qua các `LabelEncoder` đã huấn luyện trước đó, giúp mô hình dễ dàng xử lý dữ liệu. Tiếp theo, dữ liệu mã hóa này được chuyển đổi bằng kỹ thuật PCA (`bestPcaNB`), cho phép giảm số chiều dữ liệu để phù hợp với mô hình Naive Bayes. Cuối cùng, mô hình Naive Bayes (`modelMaxNB`) sử dụng dữ liệu đã được PCA chuyển đổi để thực hiện dự đoán, và kết quả được hiển thị trực tiếp trên giao diện. Trong trường hợp dữ liệu đầu vào chưa đầy đủ, hàm sẽ hiển thị một thông báo nhắc nhở người dùng nhập đầy đủ thông tin cần thiết.

Tối ưu hóa thuật toán Naive Bayes kết hợp với PCA (Phân tích Thành phần Chính) để tìm ra số thành phần tối ưu giúp mô hình đạt được hiệu suất cao nhất. Ban đầu, biến `maxRateNB` được đặt bằng 0 để lưu trữ tỷ lệ dự đoán cao nhất tìm thấy trong quá trình thực thi. Sau đó, một vòng lặp được khởi động, duyệt qua từng số thành phần khác nhau của PCA, từ 1 đến tổng số thuộc tính trong dữ liệu. Trong mỗi lần lặp, thuật toán tạo ra một đối tượng PCA mới và thiết lập số thành phần cần giữ lại là `i`, nhằm loại bỏ các thuộc tính ít quan trọng nhất trong dữ liệu.

Sau khi dữ liệu được giảm chiều, nó được chia thành các tập huấn luyện (`X\_train`, `Y\_train`) và tập kiểm tra (`X\_test`, `Y\_test`). Một mô hình Decision Tree (`TreeNB`) với tiêu chí gini được khởi tạo và huấn luyện trên tập huấn luyện để xây dựng cây phân lớp. Mô hình sau đó dự đoán trên tập kiểm tra, và tỷ lệ dự đoán đúng được tính toán. Nếu tỷ lệ này cao hơn `maxRateNB` hiện tại, thuật toán sẽ cập nhật các biến lưu trữ: `maxRateNB` sẽ giữ giá trị tỷ lệ cao nhất, ‘numComponentsNB’’ lưu số thành phần tối ưu, `bestPcaNB` lưu đối tượng PCA tốt nhất, `modelMaxNB` lưu mô hình cây quyết định tốt nhất, và `bestPredNB` lưu kết quả dự đoán tối ưu.

Kết quả cuối cùng là tìm ra số thành phần PCA và mô hình tối ưu nhất, giúp tăng độ chính xác cho dự đoán với Naive Bayes khi sử dụng PCA.

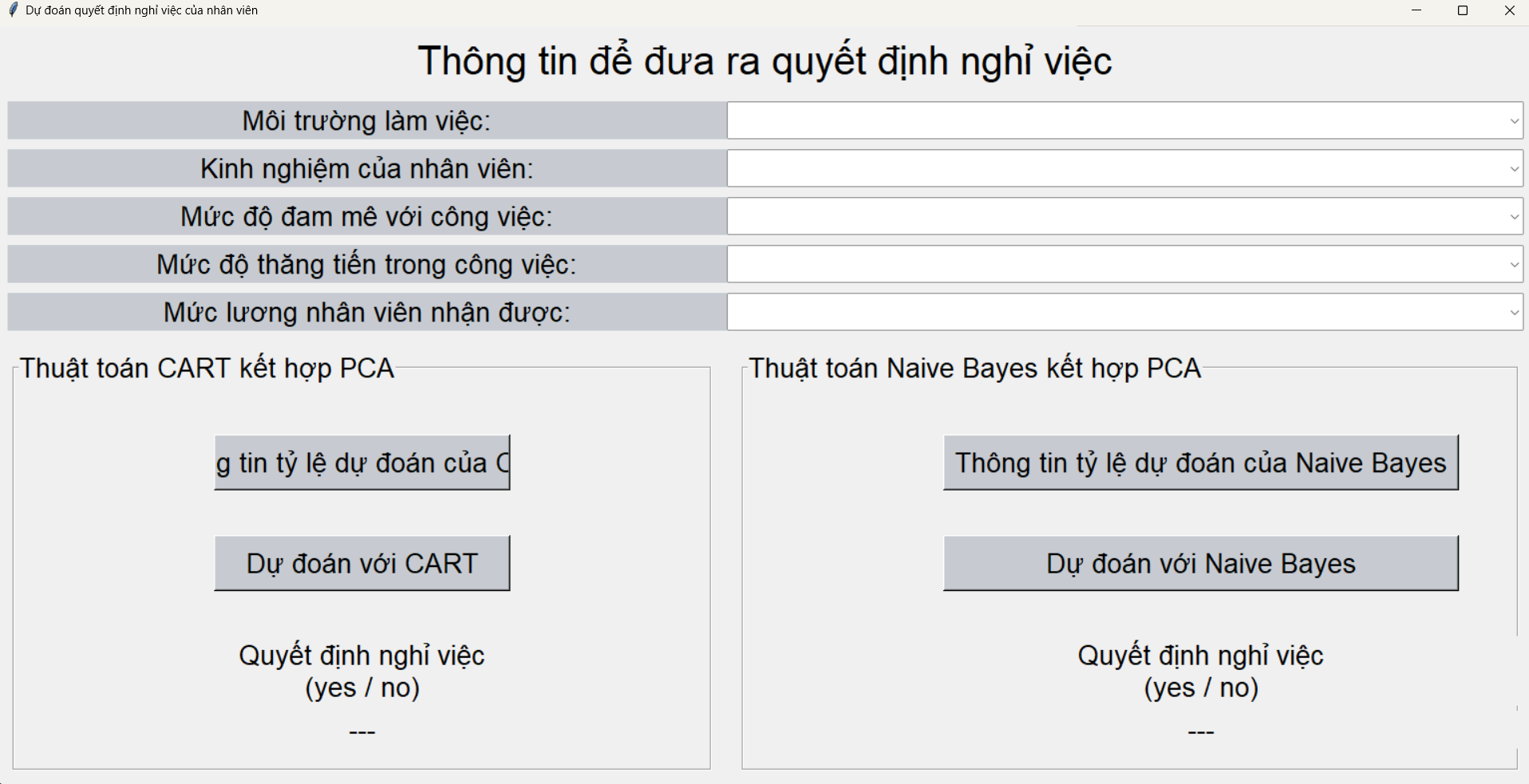
## **3.4. Đánh giá**

Hai thuật toán CART và Naive Bayes vào đề tài dự đoán quyết định nghỉ việc của nhân viên, chúng ta nhận thấy những khác biệt đáng chú ý trong hiệu suất của chúng. Thuật toán CART đạt được độ chính xác 82.5%, cho thấy khả năng phân loại vượt trội trong việc xác định các yếu tố ảnh hưởng đến quyết định nghỉ việc của nhân viên. Sự mạnh mẽ của CART đến từ khả năng xử lý các mối quan hệ phi tuyến tính và tạo ra các quy tắc phân lớp rõ ràng, giúp doanh nghiệp dễ dàng hiểu và áp dụng các quyết định dựa trên những yếu tố quan trọng.

Trong khi đó, Naive Bayes, với độ chính xác 81.25%, cũng cho thấy hiệu suất tốt trong việc dự đoán. Mặc dù thấp hơn so với CART, nhưng Naive Bayes có những lợi thế riêng, đặc biệt trong việc xử lý khối lượng dữ liệu lớn và tốc độ dự đoán nhanh. Điều này có thể rất hữu ích trong các tình huống mà thời gian phản hồi là yếu tố quan trọng, chẳng hạn như khi cần đánh giá nhanh chóng quyết định nghỉ việc dựa trên các yếu tố như mức độ hài lòng với công việc, môi trường làm việc, và cơ hội thăng tiến.

Kết quả này cho thấy rằng cả hai thuật toán đều có thể được áp dụng hiệu quả để dự đoán quyết định nghỉ việc, và sự lựa chọn giữa chúng phụ thuộc vào yêu cầu cụ thể của tổ chức. Nếu doanh nghiệp cần một mô hình dễ hiểu và giải thích rõ ràng các quyết định dựa trên dữ liệu, CART có thể là lựa chọn tốt hơn. Ngược lại, nếu cần một mô hình nhanh chóng và có khả năng xử lý tốt với khối lượng dữ liệu lớn, Naive Bayes sẽ là sự lựa chọn hợp lý.

## **3.5. Giao diện**



# **KẾT LUẬN**

Trong bài toán dự đoán quyết định nghỉ việc của nhân viên, chúng tôi đã áp dụng hai thuật toán Naive Bayes và CART, kết hợp với phương pháp phân tích thành phần chính (PCA) để giảm chiều dữ liệu. Kết quả nghiên cứu cho thấy, việc sử dụng PCA giúp tối ưu hóa hiệu suất của các mô hình, giảm bớt nhiễu trong dữ liệu và tăng độ chính xác của dự đoán. Đối với Naive Bayes, PCA đã giúp cải thiện tốc độ xử lý và tăng khả năng dự đoán đối với các dữ liệu lớn, trong khi CART kết hợp PCA đã chứng minh tính hiệu quả trong việc phân loại các đặc trưng quan trọng, giúp tăng cường tính dễ hiểu của mô hình dự đoán.

Bằng cách sử dụng các kỹ thuật này, chúng tôi đã xây dựng được một hệ thống dự đoán có độ chính xác cao trong việc xác định nguy cơ nghỉ việc của nhân viên. Đây là một bước tiến quan trọng giúp các tổ chức dự đoán và quản lý hiệu quả tình trạng nghỉ việc, từ đó có thể đưa ra các chiến lược giữ chân nhân tài phù hợp. Trong tương lai, các nghiên cứu tiếp theo có thể thử nghiệm với các thuật toán khác như Random Forest hay Gradient Boosting, hoặc mở rộng bộ dữ liệu để tăng cường khả năng dự đoán và tính tổng quát của mô hình.

Phần kết luận này nhấn mạnh sự hiệu quả của việc sử dụng Naive Bayes, CART và PCA trong dự đoán nghỉ việc, đồng thời đề xuất hướng phát triển trong tương lai để nâng cao hơn nữa hiệu quả của mô hình.

# **TÀI LIỆU THAM KHẢO**

|  |  |
| --- | --- |
| [1] | V. H. Tiệp, Machine Learning cơ bản, 2018. |
| [2] | [Trực tuyến]. Available: https://www.analyticssteps.com/blogs/classification-and-regression-tree-cart-algorithm. |
| [3] | [Trực tuyến]. Available: http://en.wikipedia.org/wiki/Naive\_Bayes\_classifier. |
| [4] | [Trực tuyến]. Available: http://en.wikipedia.org/wiki/Document\_classification. |
| [5] | [Trực tuyến]. Available: https://machinelearningcoban.com/2017/08/08/nbc/. |
| [6] | [Trực tuyến]. Available: https://www.machinelearningplus.com/predictive-modeling/how-naive-bayes-algorithm-works-with-example-and-full-code/. |
| [7] | [Trực tuyến]. Available: https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c. |

**CÔNG VIỆC MỖI NGƯỜI**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên** | **Công việc** | **Điểm** |
| 1 | Cao Thị Mai Phương | Code Naive Bayes và form |  |
| 2 | Nguyễn Đình Cường | Code CART và các chỉ tiêu đánh giá |  |
| 3 | Nguyễn Văn Luyện | Code tiền xử lý dữ liệu và viết word chương 3 |  |
| 4 | Hoàng Hiểu Nhi | Viết word chương 1, 2, kết luận |  |