TRƯỜNG ĐẠI HỌC THỦY LỢI

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP LỚN**

HỌC PHẦN: HỌC MÁY

**ĐỀ TÀI: ỨNG DỤNG THUẬT TOÁN ID3 VÀ PHÂN TÍCH THÀNH PHẦN CHÍNH (PCA) ĐỀ DỰ ĐOÁN GIÁ NHÀ Ở VIỆT NAM**

Giảng viên hướng dẫn: TS. Nguyễn Thị Kim Ngân

Nhóm sinh viên thực hiện: Nhóm 15 – lớp 60TH1

1. Lâm Văn Thái
2. Phạm Thị Phương Nga
3. Đặng Thị Thúy

**Hà Nội, năm 2021**

**MỤC LỤC**

[**PHẦN 1: TỔNG QUAN** 3](#_Toc86845549)

[***1.*** ***Giới thiệu về học máy*** 3](#_Toc86845550)

[**a.** **Lịch sử và vai trò của machine learning** 3](#_Toc86845551)

[**b.** **Ưu điểm và hạn chế của Học máy có giám sát và không giám sát** 3](#_Toc86845552)

[***2.*** ***Trình bày phương pháp học máy được sử dụng*** 4](#_Toc86845553)

[**a.** **Cây quyết đinh (Decision Tree)** 4](#_Toc86845554)

[**b.** **Phương pháp Phân tích thành phần chính (PCA)** 6](#_Toc86845555)

[***3.*** ***Bài toán*** 11](#_Toc86845556)

[**PHẦN 2: THỰC NGHIỆM** 12](#_Toc86845557)

[***1.*** ***Mô tả tập dữ liệu*** 12](#_Toc86845558)

[***2.*** ***Mô tả cách giải bài toán bằng phương pháp học máy*** 14](#_Toc86845559)

[***3.*** ***Đánh giá mô hình*** 14](#_Toc86845560)

[***4.*** ***Mô tả các chức năng của chương trình*** 15](#_Toc86845561)

[**KẾT LUẬN** 16](#_Toc86845562)

[**TÀI LIỆU THAM KHẢO** 16](#_Toc86845563)

# **PHẦN 1: TỔNG QUAN**

## ***Giới thiệu về học máy***

1. **Lịch sử và vai trò của machine learning**

* Những năm gần đây, AI - Artificial Intelligence (Trí Tuệ Nhân Tạo), và cụ thể hơn là Machine Learning (Học Máy hoặc Máy Học) nổi lên như một bằng chứng của cuộc cách mạng công nghiệp lần thứ tư (1 - động cơ hơi nước, 2 - năng lượng điện, 3 - công nghệ thông tin). Trí Tuệ Nhân Tạo đang len lỏi vào mọi lĩnh vực trong đời sống mà có thể chúng ta không nhận ra. Xe tự hành của Google và Tesla, hệ thống tự tag khuôn mặt trong ảnh của Facebook, trợ lý ảo Siri của Apple, hệ thống gợi ý sản phẩm của Amazon, hệ thống gợi ý phim của Netflix, máy chơi cờ vây AlphaGo của Google DeepMind, …, chỉ là một vài trong vô vàn những ứng dụng của AI/Machine Learning
* Machine Learning là một tập con của AI, là một lĩnh vực nhỏ của Khoa Học Máy Tính, nó có khả năng tự học hỏi dựa trên dữ liệu đưa vào mà không cần phải được lập trình cụ thể
* Những năm gần đây, khi mà khả năng tính toán của các máy tính được nâng lên một tầm cao mới và lượng dữ liệu khổng lồ được thu thập bởi các hãng công nghệ lớn, Machine Learning đã tiến thêm một bước dài và một lĩnh vực mới được ra đời gọi là Deep Learning.
* Deep Learning đã giúp máy tính thực thi những việc tưởng chừng như không thể vào 10 năm trước: phân loại cả ngàn vật thể khác nhau trong các bức ảnh, tự tạo chú thích cho ảnh, bắt chước giọng nói và chữ viết của con người, giao tiếp với con người, hay thậm chí cả sáng tác văn hay âm nhạc

1. **Ưu điểm và hạn chế của Học máy có giám sát và không giám sát**

* ***Học máy có giám sát:***
* Về cơ bản, học có giám sát là cách học mà chúng ta dạy hoặc đào tạo máy bằng cách sử dụng dữ liệu được gắn nhãn tốt, nghĩa là một số dữ liệu đã được gắn thẻ với câu trả lời đúng. Sau đó, máy được cung cấp một bộ ví dụ (dữ liệu) mới để thuật toán học có giám sát phân tích dữ liệu đào tạo (bộ ví dụ đào tạo) và tạo ra kết quả chính xác từ dữ liệu được gắn nhãn.
* *Ưu điểm:*
* Học tập có giám sát cho phép thu thập dữ liệu và tạo ra dữ liệu đầu ra từ những kinh nghiệm trước đó.
* Giúp tối ưu hóa các tiêu chí hiệu suất với sự trợ giúp của kinh nghiệm.
* ML có giám sát giúp giải quyết nhiều loại vấn đề tính toán trong thế giới thực.
* *Nhược điểm:*
* Phân loại dữ liệu lớn có thể là một thách thức.
* Đào tạo cho việc học có giám sát cần rất nhiều thời gian tính toán, vì vậy, nó đòi hỏi rất nhiều thời gian.
* ***Học không giám sát:***
* Học không giám sát là việc đào tạo máy sử dụng thông tin không được phân loại cũng như không được gắn nhãn và cho phép thuật toán hoạt động trên thông tin đó mà không cần hướng dẫn
* Các thuật toán giải quyết bài toán phân cụm và giảm chiều dữ liệu là các ví dụ điển hình của nhóm này. Trong bài toán phân cụm, có thể mô hình không trực tiếp dự đoán được đầu ra của dữ liệu nhưng vẫn có khả năng phân các điểm dữ liệu có đặc tính gần giống nhau vào từng nhóm.
* *Ưu điểm:*
* Nhóm được các thông tin chưa được sắp xếp
* không biết câu trả lời chính xác cho mỗi dữ liệu đầu vào
* *Nhược điểm:*
* Máy bị hạn chế tự tìm kiếm cấu trúc ẩn trong dữ liệu không được gắn nhãn.
* không biết câu trả lời chính xác cho mỗi dữ liệu đầu vào

## ***Trình bày phương pháp học máy được sử dụng***

1. **Cây quyết đinh (Decision Tree)**

* Cây quyết định (Decision Tree) là một cây phân cấp có cấu trúc được dùng để phân lớp các đối tượng dựa vào dãy các luật. Các thuộc tính của đối tượng có thể thuộc các kiểu dữ liệu khác nhau như Nhị phân (Binary), Định danh (Nominal), Thứ tự (Ordinal), Số lượng (Quantitative) trong khi đó thuộc tính phân lớp phải có kiểu dữ liệu là Binary hoặc Ordinal.
* ***Ý tưởng Thuật toán ID3***
* ID3 (J. R. Quinlan 1993) sử dụng phương pháp tham lam tìm kiếm từ trên xuống thông qua không gian của các nhánh có thể không có backtracking. ID3 sử dụng Entropy và Information Gain để xây dựng một cây quyết định.
* Trong ID3, chúng ta cần xác định thứ tự của thuộc tính cần được xem xét tại mỗi bước. Với các bài toán có nhiều thuộc tính và mỗi thuộc tính có nhiều giá trị khác nhau, việc tìm được nghiệm tối ưu thường là không khả thi. Thay vào đó, một phương pháp đơn giản thường được sử dụng là tại mỗi bước, một thuộc tính tốt nhất sẽ được chọn ra dựa trên một tiêu chuẩn nào đó (chúng ta sẽ bàn sớm). Với mỗi thuộc tính được chọn, ta chia dữ liệu vào các child node tương ứng với các giá trị của thuộc tính đó rồi tiếp tục áp dụng phương pháp này cho mỗi child node. Việc chọn ra thuộc tính tốt nhất ở mỗi bước như thế này được gọi là cách chọn greedy (tham lam). Cách chọn này có thể không phải là tối ưu, nhưng trực giác cho chúng ta thấy rằng cách làm này sẽ gần với cách làm tối ưu. Ngoài ra, cách làm này khiến cho bài toán cần giải quyết trở nên đơn giản hơn.
* Sau mỗi câu hỏi, dữ liệu được phân chia vào từng child node tương ứng với các câu trả lời cho câu hỏi đó. Câu hỏi ở đây chính là một thuộc tính, câu trả lời chính là giá trị của thuộc tính đó. Để đánh giá chất lượng của một cách phân chia, chúng ta cần đi tìm một phép đo.
* Trước hết, thế nào là một phép phân chia tốt? Bằng trực giác, một phép phân chia là tốt nhất nếu dữ liệu trong mỗi child node hoàn toàn thuộc vào một class – khi đó child node này có thể được coi là một leaf node, tức ta không cần phân chia thêm nữa. Nếu dữ liệu trong các child node vẫn lẫn vào nhau theo tỉ lệ lớn, ta coi rằng phép phân chia đó chưa thực sự tốt. Từ nhận xét này, ta cần có một hàm số đo độ tinh khiết (purity), hoặc độ vẩn đục (impurity) của một phép phân chia. Hàm số này sẽ cho giá trị thấp nhất nếu dữ liệu trong mỗi child node nằm trong cùng một class (tinh khiết nhất), và cho giá trị cao nếu mỗi child node có chứa dữ liệu thuộc nhiều class khác nhau.
* Một hàm số có các đặc điểm này và được dùng nhiều trong lý thuyết thông tin là hàm entropy.
* ***Entropy*** *trong Cây quyết định (Decision Tree)*
* Entropy là thuật ngữ thuộc Nhiệt động lực học, là thước đo của sự biến đổi, hỗn loạn hoặc ngẫu nhiên. Năm 1948, Shannon đã mở rộng khái niệm Entropy sang lĩnh vực nghiên cứu, thống kê với công thức như sau:
* Với một phân phối xác suất của một biến rời rạc x có thể nhận n giá trị khác nhau x1,x2,…,xn.
* Giả sử rằng xác suất để x nhận các giá trị này là pi=p(x=xi).
* Ký hiệu phân phối này là p=(p1 ,p2 ,…,pn). Entropy của phân phối này được định nghĩa là:



* ***Information Gain*** *trong Cây quyết định (Decision Tree)*
* Information Gain dựa trên sự giảm của hàm Entropy khi tập dữ liệu được phân chia trên một thuộc tính. Để xây dựng một cây quyết định, ta phải tìm tất cả thuộc tính trả về Information gain cao nhất.
* Để xác định các nút trong mô hình cây quyết định, ta thực hiện tính Information Gain tại mỗi nút theo trình tự sau:
* **Bước 1**: Tính toán hệ số Entropy của biến mục tiêu S có N phần tử với Nc phần tử thuộc lớp c cho trước:



* **Bước 2**: Tính hàm số Entropy tại mỗi thuộc tính: với thuộc tính x, các điểm dữ liệu trong S được chia ra K child node S1, S2, …, SK với số điểm trong mỗi child node lần lượt là m1, m2 ,…, mK , ta có:



* **Bước 3**: Chỉ số Gain Information được tính bằng:

**G(x, S) = H(S) – H(x,S)**

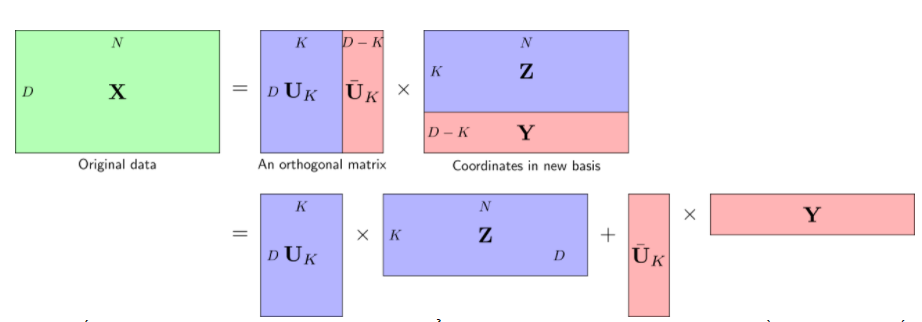
* ***Đánh giá phương pháp:***
* ID3 sử dụng một cách tiếp cận tham lam, đó là lý do tại sao nó không đảm bảo một giải pháp tối ưu; nó có thể bị mắc kẹt trong mức tối ưu cục bộ, có thể quá phù hợp với dữ liệu đào tạo (để tránh trang bị quá mức, cây quyết định nhỏ hơn nên được ưu tiên hơn cây lớn hơn). Thuật toán này thường tạo ra cây nhỏ, nhưng nó không phải lúc nào cũng tạo ra cây nhỏ nhất có thể.
* Hạn chế lớn nhất của ID3 và decision tree nói chung là việc nếu một điểm dữ liệu mới rơi vào *nhầm* nhánh ở ngay những lần phân chia đầu tiên, kết quả cuối cùng sẽ khác đi rất nhiều. Việc rơi vào nhầm nhánh này rất dễ xảy ra trong trường hợp thuộc tính liên tục được chia thành nhiều nhóm nhỏ, vì hai điểm có thuộc tính tương ứng rất gần nhau có thể rơi vào hai nhóm khác nhau.

1. **Phương pháp Phân tích thành phần chính (PCA)**

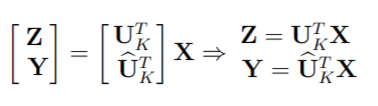
* Dimensionality Reduction (giảm chiều dữ liệu) là một trong những kỹ thuật quan trong trong nhiều bài toán, do:
* Các feature vectors trong các bài toán thực tế có thể có số chiều rất lớn tới vài nghìn; số lượng các điểm dữ liệu cũng rất lớn
* Dữ liệu có số chiều cao thì sẽ gặp khó khăn cả về lưu trữ và tốc độ tính toán

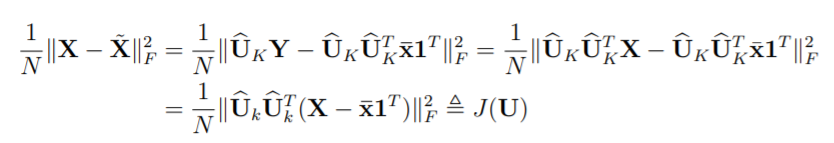
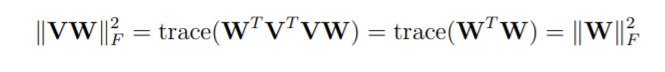
⇒ Vì vậy, giảm số chiều dữ liệu là một bước quan trọng trong nhiều bài toán. Đây cũng được coi là một phương pháp nén dữ liệu.

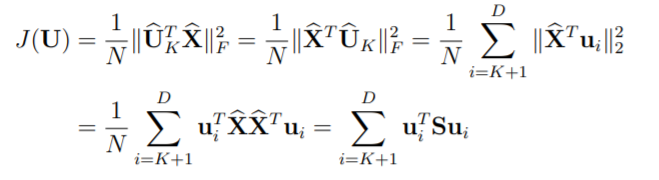
* Dimensionality Reduction, nói một cách đơn giản, là việc đi tìm một hàm số, hàm số này lấy đầu vào là một điểm dữ liệu ban đầu x ∈ RD với D rất lớn và tạo ra một điểm dữ liệu mới z ∈ RK có số chiều K < D
* Một phương pháp đơn giản nhất trong các thuật toán Dimensionality Reduction - Phương pháp Phân tích thành phần chính (Principal Component Analysis) :
* Dựa trên quan sát rằng dữ liệu thường không phân bố ngẫu nhiên trong không gian mà thường phân bố gần các đường/mặt đặc biệt nào đó
* PCA xem xét một trường hợp đặc biệt khi các mặt đặc biệt đó có dạng tuyến tính là các không gian con (subspace).
* PCA là phương pháp đi tìm một phép xoay trục tọa độ để được một hệ trục toạ độ mới sao cho trong hệ mới này, thông tin của dữ liệu chủ yếu tập trung ở một vài thành phần. Phần còn lại chứa ít thông tin hơn có thể được lược bỏ.
* Ý tưởng chính là tìm một hệ trực chuẩn mới sao cho trong hệ này, các thành phần quan trọng nhất nằm trong K thành phần đầu tiên

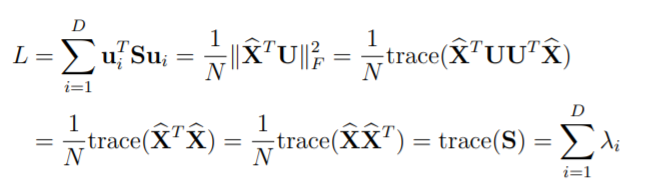


* Quan sát hình vẽ trên với cơ sở mới  là một hệ trực chuẩn với Uk là ma trận con tạo bởi K cột đầu tiên của U. Trong hệ cơ sở mới này, ma trận dữ liệu có thể được viết thành 
* Từ đây suy ra:



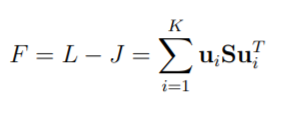
* Mục đích của PCA là đi tìm ma trận trực giao U sao cho phần lớn thông tin nằm ở UKZ, phần nhỏ thông tin nằm ở . Phần nhỏ này sẽ được lược bỏ và xấp xỉ bằng một ma trận có các cột như nhau
* ***Hàm mất mát:***
* Hàm mất mát của PCA được coi như sai số của phép xấp xỉ, được định nghĩa là 
* Chú ý rằng, nếu các cột của một ma trận V tạo thành một hệ trực chuẩn thì với một ma trận W bất kỳ, ta luôn có
* Đặt . Ma trận này có được bằng cách trừ mỗi cột của X đi trung bình các cột của nó. Ta gọi là ma trận dữ liệu đã được chuẩn hoá. Có thể thấy xˆn = xn − x¯, ∀n = 1, 2, . . . , N.
* Vì vậy ham mất mát được viết lại thành:

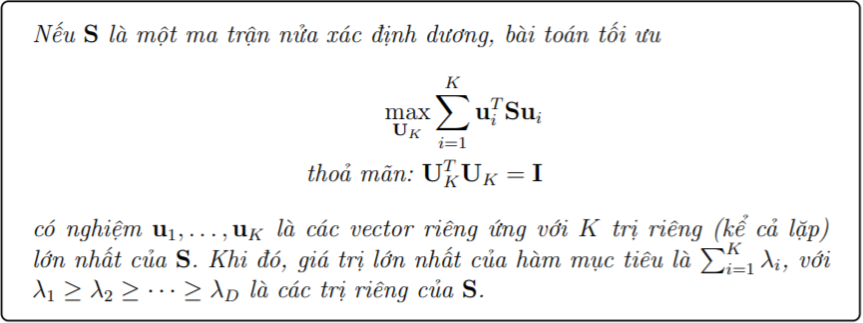
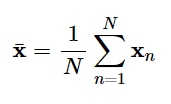


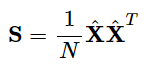
Với là ma trận hiệp phương sai của dữ liệu và luôn là một ma trận nửa xác định dương. Với U là ma trận trực giao bất kỳ, thay K = 0 vào hàm trên ta được 

Với λ1 ≥ λ2 ≥ · · · ≥ λD ≥ 0 là các trị riêng của ma trận nửa xác định dương S. Chú ý rằng các trị riêng này là thực và không âm. Như vậy L không phụ thuộc vào cách chọn ma trận trực giao U và bằng tổng các phần tử trên đường chéo của S. Nói cách khác, L chính là tổng các phương sai theo từng thành phần của dữ liệu ban đầu

Vì vậy, việc tối thiểu hàm mất mát J được cho bởi (21.9) tương đương với việc tối đa biểu thức:



* ***Tối ưu hàm mất mát:***
* Nghiệm của bài toán tối ưu hàm mất mát PCA được tìm dựa trên khẳng định sau đây
* Như đã khẳng định ở trên, tổng phương sai theo toàn bộ các chiều chiều trong một hệ cơ sở bất kỳ là như nhau và bằng tổng các trị riêng của ma trận hiệp phương sai. Vì vậy, PCA còn được coi là phương pháp giảm số chiều dữ liệu sao tổng phương sai còn lại là lớn nhất.
* Tóm tắt các bước thực hiện trong phương pháp trong PCA:
* Bước 1: Tính vector kỳ vọng của toàn bộ dữ liệu: 
* Bước 2: Trừ mỗi điểm dữ liệu đi vector kỳ vọng của toàn bộ dữ liệu: 
* Bước 3: Đặt [, , . . . , ] là ma trận dữ liệu chuẩn hoá, tính ma trận hiệp phương sai:



* Bước 4: Tính các trị riêng và vector riêng có norm bằng 1 của ma trận này, sắp xếp chúng theo thứ tự giảm dần của trị riêng.
* Bước 5: Chọn K vector riêng ứng với K trị riêng lớn nhất để xây dựng ma trận UK có các cột tạo thành một hệ trực giao. K vector này được gọi là các thành phần chính, tạo thành một không gian con gần với phân bố của dữ liệu ban đầu đã chuẩn hóa.
* Bước 6: Chiếu dữ liệu ban đầu đã chuẩn hoá xuống không gian con tìm được.
* Bước 7: Dữ liệu mới là toạ độ của các điểm dữ liệu trên không gian mới: 

*⇒ Như vậy, PCA là kết hợp của phép tịnh tiến, xoay trục tọa độ và chiếu dữ liệu lên hệ toạ độ mới*.

Dữ liệu ban đầu có thể tính được xấp xỉ theo dữ liệu mới bởi x ≈ UKZ + x¯.

* ***Đánh giá phương pháp:***
* PCA là phương pháp giảm chiều dữ liệu dựa trên việc tối đa lượng thông tin được giữ lại giúp việc lữu trữ và tính toán được thuận tiện hơn. Thực tế cho thấy nhiều khi làm việc trên dữ liệu đã đươc giảm chiều mang lại kết quả tốt hơn so với dữ liệu gốc. Việc giảm chiều dữ liệu phần nào giúp khắc phục hiện tượng gây nhiễu của dữ liệu và quá khớp khi số chiều dữ liệu quá ít. Đồng thời cải thiện trực quan hóa dữ liệu (dễ trực quan hóa khi có ít chiều)
* Lượng thông tin được giữ lại được đo bằng tổng các phương sai trên mỗi thành phần của dữ liệu. Lượng dữ liệu sẽ được giữ lại nhiều nhất khi các chiều dữ liệu còn lại tương ứng với các vector riêng của trị riêng lớn nhất của ma trận hiệp phương sai.
* Việc giảm chiều dữ liệu từ D về K < D, chỉ giữ lại K phần tử quan trọng nhất là cách đơn giản nhất. Tuy nhiên, việc làm này chắc chắn chưa phải tốt nhất vì chúng ta chưa biết xác định thành phần nào là quan trọng hơn. Hoặc trong trường hợp xấu nhất, lượng thông tin trong mỗi thành phần là như nhau, bỏ đi thành phần nào cũng dẫn đến việc mất một lượng thông tin lớn.

## ***Bài toán***

* Tên bài toán: Ứng dụng thuật toán ID3 và Phân tích thành phần chính để dự đoán giá nhà ở Việt Nam.
* Input (11 trá trị ): gồm
* BEDROOMS ( PHÒNG NGỦ )
* BATHROOMS ( PHÒNG TẮM )
* STORIES (TỒN TẠI)
* MAINROAD ( ĐƯỜNG CHÍNH )
* GUESTROOM ( PHÒNG KHÁCH )
* BASEMENT (TẦNG HẦM)
* HOTWATERHEATING (BÌNH NÓNG LẠNH)
* AIRCONDITIONING (ĐIỀU HÒA NHIỆT ĐỘ)
* PARKING (BÃI ĐẬU XE)
* PREFAREA (KHO)
* FURNITURETATUS (NỘI THẤT)
* Output: dự đoán giá nhà
* PRICE\_SEGMENT (TẦM GIÁ)

# 

# **PHẦN 2: THỰC NGHIỆM**

## ***Mô tả tập dữ liệu***

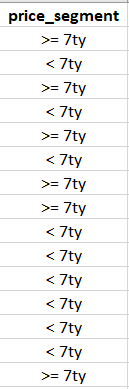
* Dữ liệu bài toán gồm: ma trận dữ liệu (X), nhãn lớp (Y)
* Ma trận x là vector đầu vào tập hợp các mẫu dữ liệu, mỗi mẫu dữ liệu là một vector mô tả giá nhà nhỏ hơn, lớn hơn hoặc bằng 7 tỷ là nhãn lớp y (giá trị thực của đầu ra) gồm có 11 thông tin dữ liệu:
* BEDROOMS ( PHÒNG NGỦ )
* BATHROOMS ( PHÒNG TẮM )
* STORIES (LỊCH SỬ LÂU ĐỜI)
* MAINROAD ( ĐƯỜNG CHÍNH )
* GUESTROOM ( PHÒNG KHÁCH )
* BASEMENT (TẦNG HẦM)
* HOTWATERHEATING (BÌNH NÓNG LẠNH)
* AIRCONDITIONING (ĐIỀU HÒA NHIỆT ĐỘ)
* PARKING (BÃI ĐẬU XE)
* PREFAREA (KHO)
* FURNITURETATUS (NỘI THẤT)
* Bài toán có 150 mẫu dữ liệu.
* Dữ liệu bài toán:

Dataset: <https://www.kaggle.com> .

* Mô tả ma trận dữ liệu (X), nhãn lớp (Y)
* Ma trận dữ liệu X: demo 15 mẫu dữ liệu ma trận X



* Ma trận nhãn lớp Y: demo 15 mẫu dữ liệu nhãn lớp Y



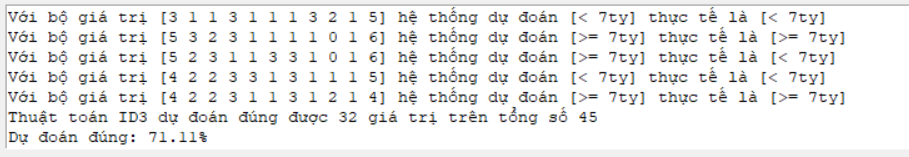
* Chia tập dữ liệu thành 2 phần: 70% dùng để huấn luyện mô hình, 30% dùng để kiểm tra sự phù hợp của mô hình.
* Tập dữ liệu có 150 mẫu dữ liệu chia: 105 mẫu dữ liệu dùng để huấn luyện mô hình, 45 mẫu dữ liệu dùng để kiểm tra sự phù hợp của mô hình.

## ***Mô tả cách giải bài toán bằng phương pháp học máy***

* Bước 1: Dùng phương pháp Phân tích thành phần chính để lựa chọn tập các thuộc tính tốt nhất cho bài toán. Từ tập training data và test data ban đầu, sử dụng các thành phần chính tốt nhất đã chọn để tạo ra tập training data và test data mới.
* Bước 2: Dùng phương pháp ID3 trong nhóm để xây dựng mô hình và đánh giá mô hình trên tập dữ liệu mới.

## ***Đánh giá mô hình***

* Kết quả thu được: Tỷ lệ dự đoán đúng sau khi tập test được phương pháp PCA lựa chọn các thuộc tính tốt nhất là 71,11 %

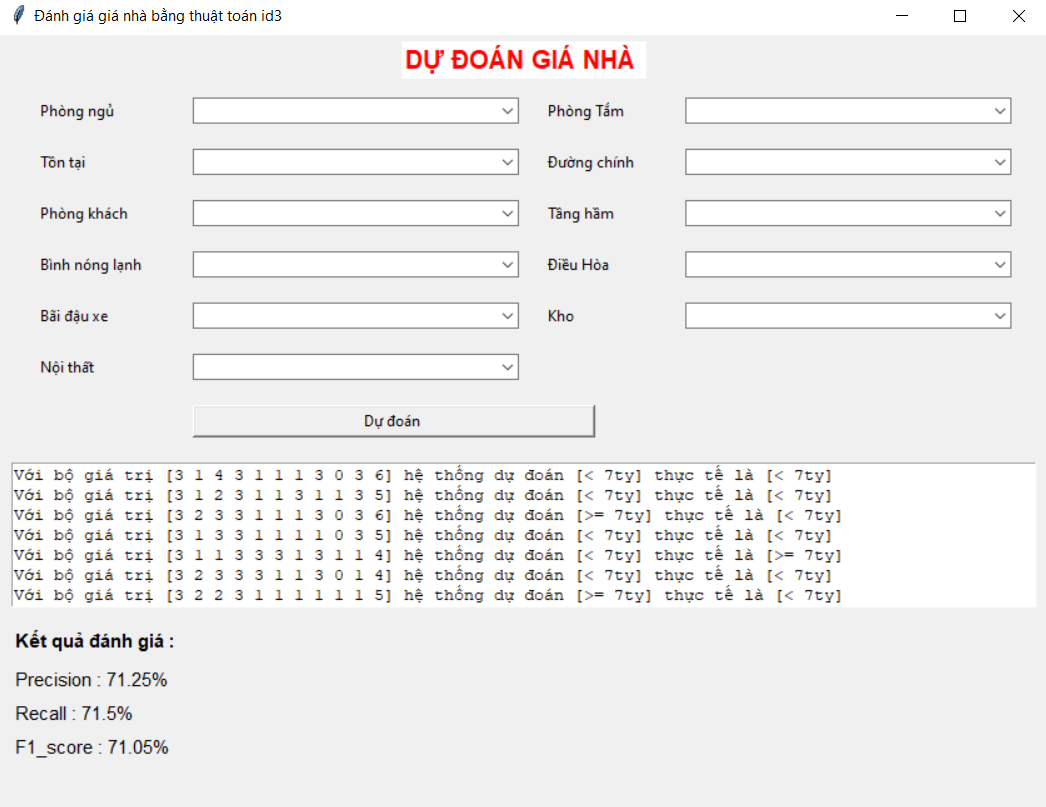


* Dùng tập test data để đánh giá chất lượng của mô hình được lựa chọn theo các độ đo:
* Precision (độ chính xác) là 71,25%
* Recall (độ nhạy) là 71,5%
* F1 - score (trung bình hài hòa của Precision và Recall) là 71,05%

- Sau khi sử dụng phương pháp PCA để giảm chiều dữ liệu và sử dựng thuật toán ID3, kết quả ta thu được khả quan hơn nhiều so với dữ liệu ban đầu thu được.

- Bằng cách sử dụng PCA trước để giảm kích thước của tập dữ liệu đào tạo đã ngăn thuật toán ID3 trang bị quá nhiều trên dữ liệu thưa thớt. Từ đó cho ra kết quả dự đoán có tỷ lệ tương đối cao

## ***Mô tả các chức năng của chương trình***

****

*Demo giao diện của chương trình*

# **KẾT LUẬN**

Các nội dung chính mà bài tập lớn làm được:

* Xây dựng được mô hình bài toán dự đoán giá nhà từ phương pháp Phân tích thành phần chính (PCA) và phương pháp ID3
* Bài toán sử dụng phương pháp Phân tích thành phần chính (PCA) lựa chọn tập các thuộc tính tốt nhất cho bài toán từ tập training data và test data để tạo ra tập training data mới và test data mới tối ưu để kết quả dự đoán được tốt hơn. Từ tập data mới đó, sử dụng vào thuật toán ID3 dự đoán được chính xác nhất giá nhà ở Việt Nam
* Tạo một giao diện nhập số liệu đầu vào, hiển thị được kết quả dự đoán, tỷ lệ dự đoán đúng đồng thời đưa ra kết quả đánh giá chương trình dựa trên các phép đo như Precision, Recall và F1-score

# **TÀI LIỆU THAM KHẢO**

* Tham khảo slide “Học máy” của cô TS. Nguyễn Thị Kim Ngân
* Tài liệu file:<https://github.com/tiepvupsu/ebookMLCB> của thầy Vũ Hữu Tiệp
* Dataset:[https://www.kaggle.com/](https://www.kaggle.com/goyalshalini93/car-price-prediction-linear-regression-rfe/data?fbclid=IwAR0x2sXrpPkU5PBKFhvSrWrJL344T-Kwyx5Qtunm_z4TOkW3xHsPJ7Slg3s)
* Tài liệu machine learning cơ bản trên internet