TRƯỜNG ĐẠI HỌC THỦY LỢI

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO BÀI TẬP LỚN MÔN HỌC**

**HỌC MÁY**

**Đề tài:**

**BÀI TOÁN DỰ ĐOÁN THUÊ PHÒNG TRỌ**

**Nhóm sinh viên thực hiện: Nhóm 19**

**Thành viên nhóm: Mai Phương Loan**

**Trần Thị Vy**

**Dương Thu Hoài**

**Giảng viên phụ trách môn học:**

**PGS.TS Nguyễn Hữu Quỳnh**

*Hà Nội, ngày 01 tháng 11 năm 2021*

Mục lục

[**Phần 1: Tổng quan** 4](#_Toc86679128)

[1. Giới thiệu về học máy 4](#_Toc86679129)

[1.1. Lịch sử và vai trò của machine learning 4](#_Toc86679130)

[1.2. Ưu điểm và hạn chế của Học có giám sát và không giám sát 6](#_Toc86679131)

[1.2.1. Supervised Learning (Học có giám sát) 6](#_Toc86679132)

[1.2.2. Unsupervised Learning (Học không giám sát) 6](#_Toc86679133)

[2. Trình bày bài toán: 7](#_Toc86679134)

[2.1. Input 7](#_Toc86679135)

[2.2. Output 8](#_Toc86679136)

[3. Phương pháp học máy được sử dụng 8](#_Toc86679137)

[3.1. Giới thiệu: 8](#_Toc86679138)

[3.2. Ưu nhược điểm của cây quyết định 9](#_Toc86679139)

[3.3. Công thức tính toán 9](#_Toc86679140)

[3.3.1. Thuật toán 9](#_Toc86679141)

[3.3.2. Tiêu chuẩn chọn thuộc tính của ID3 10](#_Toc86679142)

[3.3.3. Độ tăng thông tin IG 10](#_Toc86679143)

[3.3.4. Các đặc điểm của ID3 11](#_Toc86679144)

[3.3.5. Training error và test error 11](#_Toc86679145)

[3.3.6. Chống quá vừa dữ liệu bằng cắt tỉa cây (Overfitting) 11](#_Toc86679146)

[**Phần 2: Thực nghiệm** 12](#_Toc86679147)

[1. Mô tả tập ví dụ huấn luyện và tập test 12](#_Toc86679148)

[2. Mô tả chương trình demo 12](#_Toc86679149)

[3. Mô tả kết quả dự đoán 12](#_Toc86679150)

[4. Mô tả Chương trình demo 13](#_Toc86679151)

[4.1. Thư viện 13](#_Toc86679152)

[4.2. Đọc dữ liệu Mẫu huấn luyện, chuyển đổi dữ liệu sang dạng int 13](#_Toc86679153)

[4.3. In ra dự đoán thuê hay không thuê (Y/N) 14](#_Toc86679154)

[4.4. Đo độ chính xác của tập train 14](#_Toc86679155)

[4.5. Vẽ cây quyết định 14](#_Toc86679156)

[4.6. Xây dựng cây 15](#_Toc86679157)

[4.7. Đọc dữ liệu tập Test và chuyển đổi kiểu dữ liệu object sang kiểu dữ liệu int 15](#_Toc86679158)

[4.8. In ra cây dự đoán 16](#_Toc86679159)

[4.9 Tạo giao diện label để nhập 16](#_Toc86679160)

[4.10. Tạo giao diện lấy dữ liệu sau khi nhập và in ra dự đoán 17](#_Toc86679161)

[**Phần 3: Kết luận** 18](#_Toc86679162)

[**Tài liệu tham khảo** 19](#_Toc86679163)

# **Phần 1: Tổng quan**

## 1. Giới thiệu về học máy

### 1.1. Lịch sử và vai trò của machine learning

Trường hợp đầu tiên của mạng nơ-ron là vào năm 1943, khi nhà sinh lý học thần kinh Warren McCulloch và nhà toán học Walter Pitts viết một bài báo về các nơron và cách chúng hoạt động. Họ quyết định tạo ra một mô hình này bằng cách sử dụng một mạch điện, và do đó mạng nơ-ron ra đời.

Năm 1950 Alan Turing đã tạo ra Bài kiểm tra Turing nổi tiếng thế giới. Bài kiểm tra này khá đơn giản - để một máy tính vượt qua, nó phải thuyết phục được con người rằng đó là con người chứ không phải máy tính.

Năm 1952 chứng kiến ​​chương trình máy tính đầu tiên có thể học khi nó chạy. Đó là một trò chơi chơi cờ, được tạo ra bởi Arthur Samuel.

Frank Rosenblatt đã thiết kế mạng nơ-ron nhân tạo đầu tiên vào năm 1958, được gọi là Perceptron. Mục tiêu chính của việc này là nhận dạng mẫu và hình dạng.

Một trường hợp cực kỳ sớm khác của mạng nơ-ron là vào năm 1959, khi Bernard Widrow và Marcian Hoff tạo ra hai mô hình của chúng tại Đại học Stanford. Đầu tiên được gọi là ADELINE, và nó có thể phát hiện các mẫu nhị phân. Thế hệ tiếp theo được gọi là MADELINE, và nó có thể loại bỏ tiếng vọng trên đường dây điện thoại, vì vậy đã có một ứng dụng hữu ích trong thế giới thực.

* Những năm 1980 và 1990:

Năm 1982 là năm mà mối quan tâm đến mạng nơ-ron bắt đầu tăng trở lại, khi John Hopfield đề xuất tạo ra một mạng lưới có các đường hai chiều, tương tự như cách các nơron thực sự hoạt động. Hơn nữa, vào năm 1982, Nhật Bản tuyên bố họ đang tập trung vào các mạng nơ-ron tiên tiến hơn, nhằm khuyến khích sự tài trợ của Mỹ vào khu vực này và do đó đã tạo ra nhiều nghiên cứu hơn trong lĩnh vực này.

Mạng nơron sử dụng sự lan truyền ngược (được giải thích chi tiết trong phần Giới thiệu về Mạng thần kinh) và bước quan trọng này đến vào năm 1986, khi ba nhà nghiên cứu từ khoa tâm lý học Stanford quyết định mở rộng một thuật toán do Widrow và Hoff tạo ra vào năm 1962. Do đó, điều này cho phép nhiều các lớp được sử dụng trong mạng nơ-ron, tạo ra cái được gọi là 'người học chậm', sẽ học trong một thời gian dài.

Cuối những năm 1980 và 1990 đã không mang lại nhiều điều cho lĩnh vực này. Tuy nhiên vào năm 1997, máy tính Deep Blue của IBM, một máy tính chơi cờ vua, đã đánh bại nhà vô địch cờ vua thế giới. Kể từ đó, đã có nhiều tiến bộ hơn trong lĩnh vực này, chẳng hạn như vào năm 1998, khi nghiên cứu tại Phòng thí nghiệm AT&T Bell về nhận dạng chữ số cho kết quả chính xác tốt trong việc phát hiện mã bưu điện viết tay từ Bưu điện Hoa Kỳ. Sự lan truyền ngược được sử dụng này, như đã nêu ở trên, được giải thích chi tiết trong phần Giới thiệu về Mạng thần kinh.

* Thế kỷ 21:

Kể từ đầu thế kỷ 21, nhiều doanh nghiệp đã nhận ra rằng học máy sẽ làm tăng tiềm năng tính toán. Đây là lý do tại sao họ đang nghiên cứu nhiều hơn về nó, để đi trước đối thủ.

Một số dự án lớn bao gồm:

Google Brain (2012) - Đây là một mạng thần kinh sâu do Jeff Dean của Google tạo ra, tập trung vào việc phát hiện mẫu trong hình ảnh và video. Nó có thể sử dụng các tài nguyên của Google, điều này khiến nó không thể so sánh được với các mạng nơ-ron nhỏ hơn nhiều. Sau đó, nó được sử dụng để phát hiện các đối tượng trong video YouTube.

AlexNet (2012) - AlexNet đã giành chiến thắng trong cuộc thi ImageNet với tỷ suất lợi nhuận lớn vào năm 2012, dẫn đến việc sử dụng GPU và Mạng thần kinh hợp pháp trong học máy. Họ cũng tạo ra ReLU, đây là một chức năng kích hoạt giúp cải thiện đáng kể hiệu quả của CNN.

Có hai cách phổ biến phân nhóm các thuật toán Machine learning. Một là dựa trên phương thức học (learning style), hai là dựa trên chức năng (function).

Theo phương thức học, các thuật toán Machine Learning thường được chia làm 4 nhóm: Supervised learning, Unsupervised learning, Semi-supervised learning và Reinforcement learning.Ở đây nhóm em chỉ đề cập tới Supervised Learning và Unsupervised Learning.

### 1.2. Ưu điểm và hạn chế của Học có giám sát và không giám sát

#### 1.2.1. Supervised Learning (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.

Một cách toán học, Supervised learning là khi chúng ra có một tập hợp biến đầu vào X={x\_1,x\_2,…,x\_n } và một tập hợp nhãn tương ứng Y={y\_1,y\_2,…,y\_n }, trong đó x\_i, y\_i là các vector.Các cặp dữ liệu biết trước (x\_i, y\_i) ∈X ×Y được gọi là tập training data (dữ liệu huấn luyện). Từ tập training data này, chúng ta cần tạo ra một hàm số ánh xạ mỗi phần tử từ tập X sang một phần tử (xấp xỉ) tương ứng của tập Y:

y\_i≈f(x\_i ), ∀i=1,2,…,N

Mục đích là xấp xỉ hàm số f, thật tốt để khi có một dữ liệu x mới, chúng ta có thể tính được nhãn tương ứng của nó y=f(x).

* Ưu điểm:
  + Giúp tối ưu hóa các tiêu chí hiệu suất và giải quyết các loại vấn đề tính toán bằng cách sử dụng kinh nghiệm.
  + Tính rõ ràng của dữ liệu (làm việc trên dữ liệu được gắn nhãn và do đó dễ phân loại và dễ đào tạo).
* 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.
  + Dữ liệu không mong muốn có thể làm giảm độ chính xác

Thuật toán supervised learning được chia ra thành hai loại chính: Classification (Phân loại) và Regression (Hồi quy).

#### 1.2.2. Unsupervised Learning (Học không giám sát)

Trong thuật toán này, chúng ta không biết được outcome hay nhãn mà chỉ có dữ liệu đầu vào. 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.

Một cách toán học, Unsupervised learning là khi chúng ta chỉ có dữ liệu vào X mà không biết nhãn Y tương ứng.

Các bài toán Unsupervised learning được chia thành hai loại: Clustering (phân nhóm) và Association.

Ưu điểm :

Mô hình học máy không giám sát tìm thấy tất cả các loại mẫu chưa biết trong dữ liệu . Do đó, nó có thể giúp bạn phát hiện các tính năng có thể hữu ích trong việc phân loại dữ liệu.

Hơn nữa, trong mô hình học tập không giám sát, không cần phải gắn nhãn các đầu vào dữ liệu. Và dữ liệu không có nhãn, nhìn chung, dễ lấy hơn, vì nó có thể được lấy trực tiếp từ máy tính mà không cần thêm sự can thiệp của con người. Điều này làm cho việc học không có giám sát trở thành một mô hình ít phức tạp hơn so với các kỹ thuật học có giám sát.

Nhược điểm :

Học không giám sát về bản chất khó hơn học có giám sát vì nó không có đầu ra tương ứng.

Kết quả của thuật toán học tập không được giám sát có thể kém chính xác hơn vì dữ liệu đầu vào không được gắn nhãn và các thuật toán không biết trước kết quả đầu ra chính xác.

## 2. Trình bày bài toán:

### 2.1. Input

Bài toán được dựa trên thông tin thực tế đã qua khảo sát, gồm 5 tập biến đầu vào sau:

- Khoảng cách đến trường học (được tính theo đơn vị km)

- Diện tích (được tính theo đơn vị m2)

- Giá tiền (được tính theo đơn vị triệu đồng)

- Khép kín hay không (gồm 2 giá trị: có, không)

- Thông thoáng hay không (gồm 2 giá trị: có, không)

### 2.2. Output

Có nên thuê phòng trọ sau khi đã dựa theo và phân tích những số liệu, yếu tố đã nêu ra hay không (gồm 2 giá trị: có và không)

## 3. Phương pháp học máy được sử dụng

### 3.1. Giới thiệu:

Trong lĩnh vực máy học, cây quyết định là một kiểu mô hình dự báo (predictive model), nghĩa là một ánh xạ từ các quan sát về một sự vật/hiện tượng tới các kết luận về giá trị mục tiêu của sự vật/hiện tượng.

Decision tree là một mô hình supervised learning, có thể được áp dụng vào cả hai bài toán classification và regression. Việc xây dựng một decision tree trên dữ liệu huấn luyện cho trước là việc đi xác định các câu hỏi và thứ tự của chúng.

Ý tưởng :

* chúng ta cần xác định thứ tự của thuộc tính cần được xem xét tại mỗi bước.
* tại mỗi bước, một thuộc tính tốt nhất sẽ được chọn ra dựa trên một tiêu chuẩn nào đó.
* 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.

Mỗi một nút trong (internal node) tương ứng với một biến. Có 3 loại nút:

* Nút gốc
* Nút nội bộ: mang tên thuộc tính của CSDL
* Nút lá: mang tên lớp C. Mỗi nút lá đại diện cho giá trị dự đoán của biến mục tiêu, cho trước các giá trị của các biến được biểu diễn bởi đường đi từ nút gốc tới nút lá đó.

Các nhánh: là đường nối giữa nó với nút con của nó thể hiện một giá trị cụ thể cho biến đó.

Kỹ thuật học máy dùng trong cây quyết định được gọi là học bằng cây quyết định, hay chỉ gọi với cái tên ngắn gọn là cây quyết định.

### 3.2. Ưu nhược điểm của cây quyết định

+) So với các phương pháp khai phá dữ liệu khác, cây quyết định là phương pháp có một số ưu điểm:

* Cây quyết định dễ hiểu. Người ta có thể hiểu mô hình cây quyết định sau khi được giải thích ngắn.
* Việc chuẩn bị dữ liệu cho một cây quyết định là cơ bản hoặc không cần thiết. Các kỹ thuật khác thường đòi hỏi chuẩn hóa dữ liệu, cần tạo các biến phụ (dummy variable) và loại bỏ các giá trị rỗng.
* Cây quyết định có thể xử lý cả dữ liệu có giá trị bằng số và dữ liệu có giá trị là tên thể loại. Các kỹ thuật khác thường chuyên để phân tích các bộ dữ liệu chỉ gồm một loại biến. Chẳng hạn, các luật quan hệ chỉ có thể dùng cho các biến tên, trong khi mạng nơ-ron chỉ có thể dùng cho các biến có giá trị bằng số.
* Cây quyết định là một mô hình hộp trắng. Nếu có thể quan sát một tình huống cho trước trong một mô hình, thì có thể dễ dàng giải thích điều kiện đó bằng logic Boolean. Mạng nơ-ron là một ví dụ về mô hình hộp đen, do lời giải thích cho kết quả quá phức tạp để có thể hiểu được.
* Có thể thẩm định một mô hình bằng các kiểm tra thống kê. Điều này làm cho ta có thể tin tưởng vào mô hình.
* Cây quyết định có thể xử lý tốt một lượng dữ liệu lớn trong thời gian ngắn. Có thể dùng máy tính cá nhân để phân tích các lượng dữ liệu lớn trong một thời gian đủ ngắn để cho phép các nhà chiến lược đưa ra quyết định dựa trên phân tích của cây quyết định.

+) Nhược điểm của cây quyết định - khó giải quyết được những vấn đề có dữ liệu phụ thuộc thời gian liên tục - dễ xảy ra lỗi khi có quá nhiều lớp chi phí tính toán để xây dựng mô hình cây quyết định CAO

### 3.3. Công thức tính toán

Cây quyết định được học (xây dựng) từ dữ liệu huấn luyện. Với mỗi bộ dữ liệu có thể xây dựng nhiều cây quyết định. Quá trình học là quá trình tìm kiếm cây quyết định phù hợp với dữ liệu huấn luyện.

#### 3.3.1. Thuật toán

* Khởi đầu: nút hiện thời là nút gốc chứa toàn bộ tập dữ liệu huấn luyện
* Tại nút hiện thời n, lựa chọn thuộc tính:
  + Chưa được sử dụng ở nút tổ tiên.
  + Cho phép phân chia tập dữ liệu hiện thời thành các tập con một cách tốt nhất.
  + Với mỗi giá trị thuộc tính được chọn thêm một nút con bên dưới.
  + Chia các ví dụ ở nút hiện thời về các nút con theo giá trị thuộc tính được chọn.
* Lặp (đệ quy) cho tới khi:
  + Tất cả các thuộc tỉnh đã được sử dụng ở các nút phía trên, hoặc
  + Tất cả ví dụ tại nút hiện thời có cùng nhãn phần loại.
  + Nhãn của nút được lấy theo đa số nhãn của ví dụ tại nút hiện thời.

#### 3.3.2. Tiêu chuẩn chọn thuộc tính của ID3

* Tại mỗi nút n
  + Tập (con) dữ liệu ứng với nút đó.
  + Cần lựa chọn thuộc tính cho phép phân chia tập dữ liệu tốt nhất.
* Tiêu chuẩn:
  + Dữ liệu sau khi phân chia càng đồng nhất càng tốt
  + Đo bằng độ tăng thông tin (Information Gain - IG)
  + Chọn thuộc tính có độ tăng thông tin lớn nhất
  + IG dựa trên entropy của tập (con) dữ liệu Entropy
* Trường hợp tập dữ liệu s có 2 loại nhãn: đúng (+) hoặc sai (-)

Entropy(S) = - p+ log2 p+ -p\_log2p\_

Với: p+: % số mẫu đúng

p\_: % số mẫu sai

* Trường hợp tổng quát: có C loại nhãn

Entropy(S) = ∑ci=1 -pilog2pi

Với: pi: % ví dụ của S thuộc loại i

#### 3.3.3. Độ tăng thông tin IG

Với tập (con) mẫu *S* và thuộc tính *A*

*IG(S,A)= Entropy(S) - ∑\_(V∈values(A)).|S៴ | / |S|.Entropy(S៴)*

Trong đó:

* value(A): tập các giá trị của A
* Sv là tập con của S bao gồm các mẫu có giá trị của A bằng v
* |S| số phần từ của S

#### 3.3.4. Các đặc điểm của ID3

* ID3 là thuật toán tìm kiếm cây quyết định phù hợp với dữ liệu huấn luyện
* Tìm kiếm theo kiểu tham lam, bắt đầu từ cây rỗng
* Hàm đánh giá là độ tăng thông tin
* ID3 có khuynh hướng (bias) lựa chọn cây đơn giản
  + Ít nút
  + Các thuộc tính có độ tăng thông tin lớn nằm gần gốc

#### 3.3.5. Training error và test error

* **Training error (lỗi huấn luyện)**
  + Là lỗi đo được trên tập dữ liệu huấn luyện
  + Thường đo bằng sự sai khác giữa giá trị tính toán của mô hình và giá trị thực của dữ liệu huấn luyện
  + Trong quá trình học ta cố gắng làm giảm tới mức tối thiểu lỗi huấn

luyện

* Test error (lỗi kiểm tra)
  + Là lỗi đo được trên tập dữ liệu kiểm tra
  + Là cái ta thực sự quan tâm

#### 3.3.6. Chống quá vừa dữ liệu bằng cắt tỉa cây (Overfitting)

* Chia dữ liệu thành hai phần:
  + Huấn luyện
  + Kiểm tra
* Tạo cây đủ lớn trên dữ liệu huấn luyện
* Tính độ chính xác của cây trên tập kiểm tra
* Loại bỏ cây con sao cho kết quả trên dữ liệu kiểm tra được cải thiện nhất
* Lặp lại cho đến khi không còn cải thiện được kết quả nữa

# **Phần 2: Thực nghiệm**

## 1. Mô tả tập ví dụ huấn luyện và tập test

**-** Mô tả tập huấn luyện: các yếu tố liên quan đến việc thuê nhà trọ

- Tập huấn luyện có 100 mẫu với 6 thuộc tính:

* Khoảng cách (km)
* Diện tích (m²)
* Thông thoáng
* Khép kín
* Giá Thuê (triệu)
* Thuê

- Tập test có 20 mẫu có 5 mẫu:

* Khoảng cách (km)
* Diện tích (m²)
* Thông thoáng
* Khép kín
* Giá Thuê (triệu)

- Số cột của X vào là 5

## 2. Mô tả chương trình demo

-Mô tả các chức năng của chương trình

-Chương trình bao gồm các giao diện đồ họa

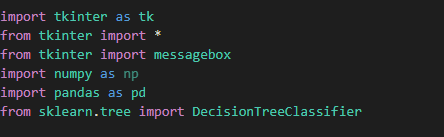
## 3. Mô tả kết quả dự đoán

Mô tả độ chính xác của phương pháp là bn %

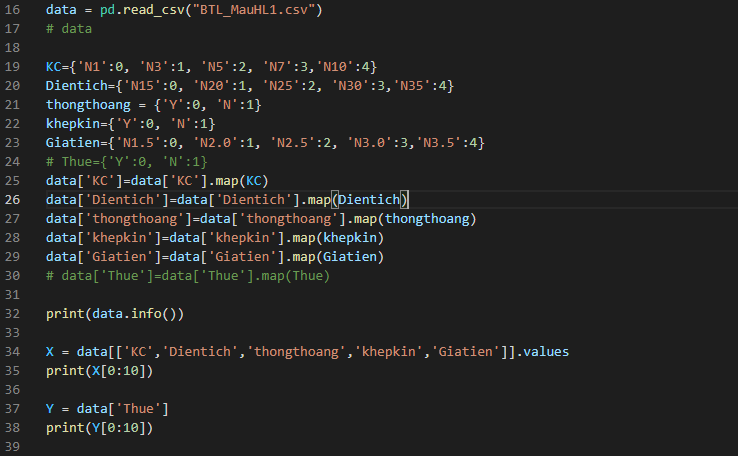
-Sử dụng độ đo accuracy (tham khảo trang machinelearningcoban)

## 4. Mô tả Chương trình demo

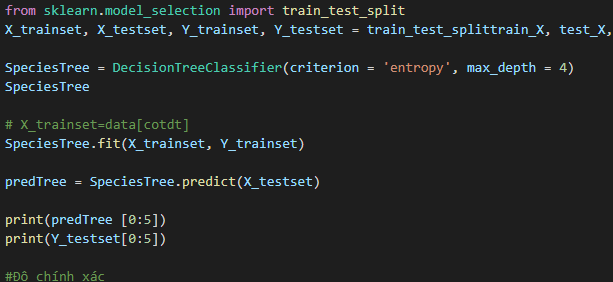
### 4.1. Thư viện

****

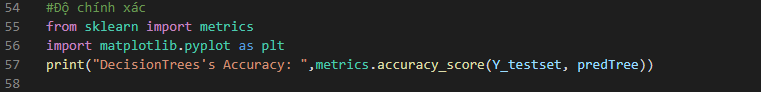
### 4.2. Đọc dữ liệu Mẫu huấn luyện, chuyển đổi dữ liệu sang dạng int



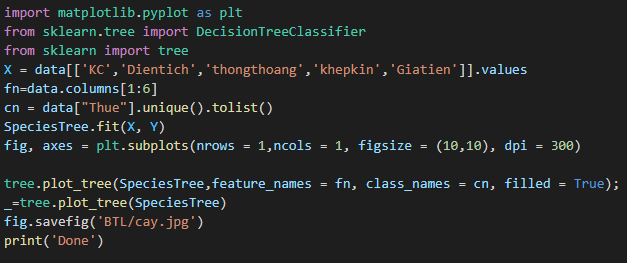
### 4.3. In ra dự đoán thuê hay không thuê (Y/N)



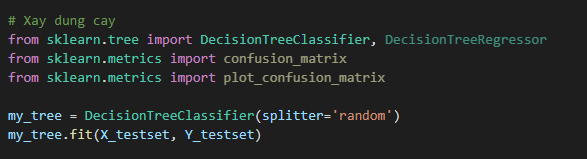
### 4.4. Đo độ chính xác của tập train



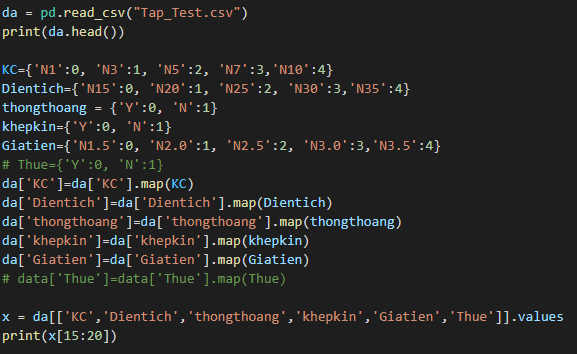
### 4.5. Vẽ cây quyết định



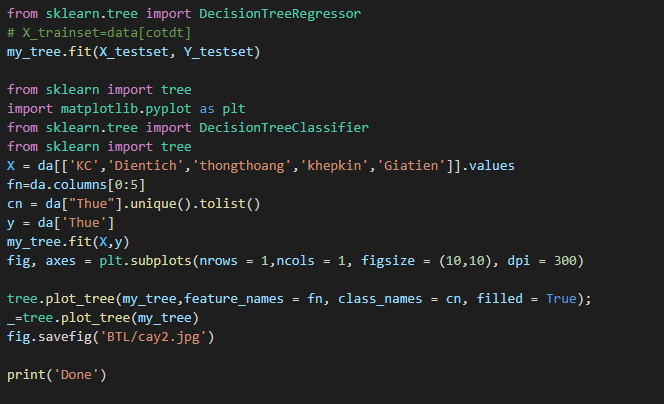
### 4.6. Xây dựng cây



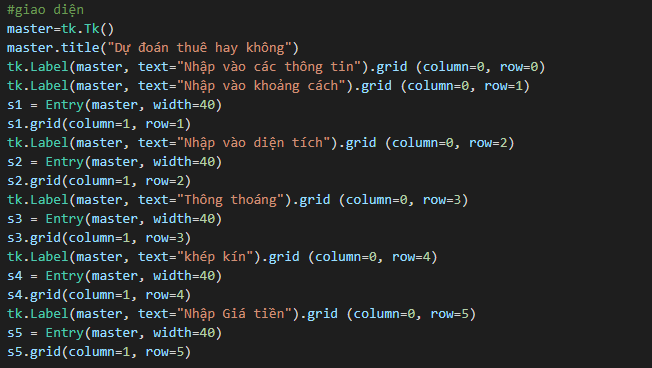
### 4.7. Đọc dữ liệu tập Test và chuyển đổi kiểu dữ liệu object sang kiểu dữ liệu int



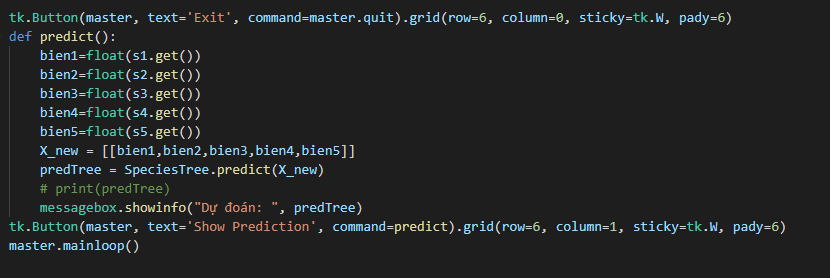
### 4.8. In ra cây dự đoán



### 4.9 Tạo giao diện label để nhập



### 4.10. Tạo giao diện lấy dữ liệu sau khi nhập và in ra dự đoán



# **Phần 3: Kết luận**

Qua quá trình học tập và nghiên cứu môn học máy cùng với sự giúp đỡ của thầy PGS.TS Nguyễn Hữu Quỳnh, nhóm đã hoàn thành đề tài môn học máy - “Dự đoán thuê nhà trọ”.

Bài tập lớn thu được những nội dung sau:

- Trang bị cho các thành viên kiến thức về lịch sử của học máy, phương pháp học có giám sát và học không giám sát.

- Nắm bắt được cách tính toán, tạo lập mẫu huấn luyện, tập test và viết chương trình code ngôn ngữ Python cho phương pháp học cây quyết định.

- Rèn luyện cho sinh viên kỹ năng làm việc theo nhóm.

Do thời gian và kinh nghiệm hạn chế nên bài có thể không đạt được tới sự hoàn hảo như đã kỳ vọng. Nhóm rất mong nhận được sự đánh giá và góp ý từ phía thầy cô để có thể rút kinh nghiệm và hoàn thiện hơn.

# **Tài liệu tham khảo**

1. Slide bài giảng của PGS.TS Nguyễn Hữu Quỳnh.
2. Hướng dẫn của thầy Nguyễn Danh Dũng: <https://www.tutorialspoint.com/python/python_gui_programming.htm>
3. <https://machinelearningcoban.com>
4. <https://www.w3schools.com/python/>
5. <https://helpex.vn/article/viduguipythonhuongdantkinter5c54d2e6507419248c9afa16>