**TRƯỜNG ĐẠI HỌC LAO ĐỘNG VÀ XÃ HỘI**



**BÁO CÁO TIỂU LUẬN**

**Học phần: Hệ điều hành**

***Đề tài*: Tìm hiểu hệ điều hành UNIX**

Họ và tên : 132- Lê Thị Huyền Trang

091- Phạm Kim Ngân

115- Bùi Công Thành

Lớp tín chỉ : D17CN04

**LỜI CẢM ƠN**

Với lòng biết ơn sâu sắc và tình cảm chân thành cho phép em gửi lời cảm ơn chân thành nhất tới:

– Trường Đại học Lao Động và Xã Hội , khoa công nghệ thông tin cùng các giảng viên đã tận tình chỉ dạy và tạo điều kiện giúp đỡ em trong quá trình học tập, nghiên cứu và hoàn thành đề tài nghiên cứu khoa học.

– Đặc biệt tôi xin bày tỏ lòng biết ơn sâu sắc đến cô Tạ Tường Vi – người hướng dẫn và cũng là người đã luôn tận tình hướng dẫn, chỉ bảo, giúp đỡ và động viên em trong suốt quá trình nghiên cứu và hoàn thành đề tài nghiên cứu này.

– Cảm ơn gia đình, bạn bè và đồng nghiệp đã luôn khích lệ, động viên và giúp đỡ tôi trong quá trình học tập và nghiên cứu khoa học.

Mặc dù đã cố gắng rất nhiều, nhưng bài luận không tránh khỏi những thiếu sót; tác giả rất mong nhận được sự thông cảm, chỉ dẫn, giúp đỡ và đóng góp ý kiến của các nhà khoa học, của quý thầy cô, các cán bộ quản lý và các bạn đồng nghiệp.

Xin chân thành cảm ơn!

**MỤC LỤC**

[MỞ ĐẦU 1](#_Toc131364776)

[I. TỔNG QUAN HỆ ĐIỀU HÀNH UNIX 3](#_Toc131364777)

[1 Hệ điều hành UNIX 3](#_Toc131364778)

[2. Các đặc điểm cơ bản 5](#_Toc131364779)

II. LỆNH VÀ TIỆN ÍCH CƠ BẢN

1.[Các lệnh khởi tạo 7](#_Toc131364781)

2. Các lệnh hiển thị

3. định hướng vào ra

[4. Desktop 8](#_Toc131364784)

5. Các lệnh thao thư mục và các tác file

6. In ấn

7. Thư tín

8. Quản lí tiến trình

9. Kiểm soát quyền hạn và bảo mật [9](#_Toc131364785)

[10. Lưu trữ và phực hồi dữ liệu](#_Toc131364786)

[11. Các thao tác trên mạng 10](#_Toc131364786)

III.THÂM NHẬP HẸ THỐNG- CÁC LỆNH CĂN BẢN ….

1. Bắt đầu và kết thúc phiên làm việc -Xác lập môi trường hệ thống

2. Các lệnh hiển thị

3. Định hướng vào ra và đường ống

4. Destop:

5. Các lệnh thao tác trên thư mục , flie

6. In ấn

7. Thư tín điện tử

8. Quản lí tiến trình

9. Các lệnh liên quan bảo quản mật và quyền hạn

a) Khái niệm:

b) Các lệnh

10. Lưu trữ và hồi phục dữ liệu

11. Các thao tác trên mạng

IV. Lập trình Shell

[TÀI LIỆU THAO KHẢO 42](#_Toc131364802)

# MỞ ĐẦU

Phương pháp học máy đóng vai trò quan trọng trong việc xây dựng các mô hình dự đoán và phân loại, từ đó giúp giải quyết các vấn đề thực tiễn. Trong tiểu luận này, chúng ta sẽ tìm hiểu về các phương pháp học máy phổ biến như: Học có giám sát (Supervised Learning), Học không giám sát (Unsupervised Learning) và Học tăng cường (Reinforcement Learning). Ngoài ra, cũng sẽ đề cập đến các vấn đề liên quan đến việc lựa chọn và đánh giá các mô hình học máy, bao gồm các phương pháp chia tập dữ liệu (Data Splitting), kiểm định chéo (Cross-Validation), và các phương pháp đánh giá kết quả dự đoán (Evaluation Metrics).

Bằng việc nghiên cứu và áp dụng các phương pháp học máy, chúng ta có thể tạo ra các mô hình học máy độ chính xác cao, giúp giải quyết các vấn đề khó khăn trong thực tế và đưa ra các quyết định thông minh, hỗ trợ cho sự phát triển của các lĩnh vực khác nhau.

Trước khi bắt đầu áp dụng các phương pháp học máy, chúng ta cần có các bước chuẩn bị dữ liệu (Data Preparation) để đảm bảo dữ liệu đầu vào đủ tốt và đáng tin cậy để huấn luyện các mô hình học máy. Các bước chuẩn bị dữ liệu bao gồm: thu thập dữ liệu, làm sạch dữ liệu (Data Cleaning), biến đổi dữ liệu (Data Transformation), và rút trích đặc trưng (Feature Extraction).

Các phương pháp học máy được chia thành ba nhóm chính: Học có giám sát, Học không giám sát và Học tăng cường. Học có giám sát được sử dụng để xây dựng các mô hình dự đoán và phân loại, dựa trên các tập dữ liệu có nhãn. Các phương pháp trong Học không giám sát được sử dụng để phân tích cấu trúc của dữ liệu, tìm kiếm các mẫu và nhóm tương đồng. Trong khi đó, Học tăng cường là một phương pháp học máy đặc biệt, nơi mô hình học tương tác với môi trường để đạt được mục tiêu được đề ra.

Việc lựa chọn và đánh giá các mô hình học máy là rất quan trọng trong quá trình xây dựng các ứng dụng học máy. Các phương pháp chia tập dữ liệu, kiểm định chéo và các phương pháp đánh giá kết quả dự đoán đóng vai trò quan trọng trong việc đánh giá và so sánh các mô hình học máy. Trong tiểu luận này, chúng ta sẽ tìm hiểu kỹ hơn về các phương pháp và ứng dụng của các phương pháp học máy cũng như các kỹ thuật liên quan để đạt được kết quả tốt nhất trong việc áp dụng học máy vào thực tiễn.

Ngoài ra , chúng ta còn tìm hiểu và xây dựng mô hình bằng phương pháp Random forest. Trong năm 2017, ta có thể chứng kiến một phong trào sôi nổi liên quan đến đề tài “Xây dựng mô hình học theo phương pháp thống kê - Statistical learning” trong cộng đồng R. Hàng loạt package mới được phát triển nhằm diễn giải cho từng algorithm chuyên biệt như hồi quy tuyến tính (GLM), Random Forest (RF) và Extreme Gradient boosting (XGB). Do phương pháp Statistical learning (tức Machine learning) ngày càng phổ biến trong nghiên cứu y học, nhu cầu diễn giải các mô hình Machine learning trở thành nhu cầu thiết yếu. Do đó, Nhi sẽ lần lượt chuyển đến các bạ hướng dẫn sử dụng những packages mới này. Bài đầu tiên này sẽ là package « randomForestExplainer » , chuyên dụng cho mô hình Random Forest.

Đây là một package vừa được công bố vào cuối tháng 7 năm 2017 bởi tác giả Aleksandra Paluszyńska. Công dụng của package này cho phép khảo sát nội dung bên trong một mô hình Random Forest.

Như chúng ta biết, Random Forest là một tập hợp mô hình (ensemble). Mô hình Random Forest rất hiệu quả cho các bài toán phân loại vì nó huy động cùng lúc hàng trăm mô hình nhỏ hơn bên trong với quy luật khác nhau để đưa ra quyết định cuối cùng. Mỗi mô hình con có thể mạnh yếu khác nhau, nhưng theo nguyên tắc « wisdom of the crowd », ta sẽ có cơ hội phân loại chính xác hơn so với khi sử dụng bất kì một mô hình đơn lẻ nào.

Như tên gọi của nó, Random Forest (RF) dựa trên cơ sở :

Random = Tính ngẫu nhiên ;

Forest = nhiều cây quyết định (decision tree).

Phương pháp RF tạo ra N các cây quyết định. Mỗi cây quyết định được tạo ra một cách ngẫu nhiên từ việc : Tái chọn mẫu (bootstrap, random sampling) và chỉ dùng một phần nhỏ tập biến ngẫu nhiên (random features) từ toàn bộ các biến trong dữ liệu. Ở trạng thái sau cùng, mô hình RF thường hoạt động rất chính xác, nhưng đổi lại, ta không thể nào hiểu được cơ chế hoạt động bên trong mô hình vì cấu trúc quá phức tạp. RF do đó là một trong số những mô hình hộp đen (black box).

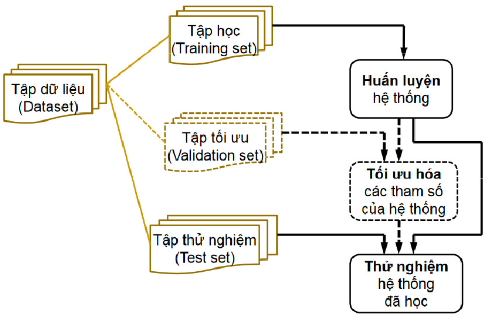
Trong quá khứ, chúng ta thường chấp nhận đánh đổi tính tường minh để đạt được tính chính xác. Từ mô hình Random Forest, chúng ta chỉ có thể làm một số khảo sát hạn chế, bao

gồm vai trò tương đối của các biến (features) và vẽ các biểu đồ 2 chiều thể hiện ranh giới các vùng phân loại.

# CHƯƠNG 1. TỔNG QUAN VỀ PHƯƠNG PHÁP HỌC MÁY

* 1. **Học máy là gì?**

Học máy ([tiếng Anh](https://vi.wikipedia.org/wiki/Ti%E1%BA%BFng_Anh): machine learning) là một lĩnh vực của [trí tuệ nhân tạo](https://vi.wikipedia.org/wiki/Tr%C3%AD_tu%E1%BB%87_nh%C3%A2n_t%E1%BA%A1o) liên quan đến việc nghiên cứu và xây dựng các kĩ thuật cho phép các hệ thống "học" tự động từ dữ liệu để giải quyết những vấn đề cụ thể. Ví dụ như các máy có thể "học" cách phân loại [thư điện tử](https://vi.wikipedia.org/wiki/Th%C6%B0_%C4%91i%E1%BB%87n_t%E1%BB%AD) xem có phải [thư rác (spam)](https://vi.wikipedia.org/wiki/Th%C6%B0_r%C3%A1c) hay không và tự động xếp thư vào thư mục tương ứng. Học máy rất gần với [suy diễn thống kê](https://vi.wikipedia.org/wiki/Suy_di%E1%BB%85n_th%E1%BB%91ng_k%C3%AA) (statistical inference) tuy có khác nhau về thuật ngữ. Một nhánh của học máy là [học sâu](https://vi.wikipedia.org/wiki/H%E1%BB%8Dc_s%C3%A2u)phát triển rất mạnh mẽ gần đây và có những kết quả vượt trội so với các phương pháp học máy khác Học máy có liên quan lớn đến [thống kê](https://vi.wikipedia.org/wiki/Khoa_h%E1%BB%8Dc_Th%E1%BB%91ng_k%C3%AA), vì cả hai lĩnh vực đều nghiên cứu việc phân tích dữ liệu, nhưng khác với thống kê, học máy tập trung vào sự phức tạp của các giải thuật trong việc thực thi tính toán. Nhiều bài toán suy luận được xếp vào loại bài toán [NP-khó](https://vi.wikipedia.org/wiki/NP-kh%C3%B3), vì thế một phần của học máy là nghiên cứu sự phát triển các giải thuật suy luận xấp xỉ mà có thể xử lý được.



Học máy có hiện nay được áp dụng rộng rãi bao gồm [máy truy tìm dữ liệu](https://vi.wikipedia.org/wiki/M%C3%A1y_truy_t%C3%ACm_d%E1%BB%AF_li%E1%BB%87u), [chẩn đoán y khoa](https://vi.wikipedia.org/wiki/Ch%E1%BA%A9n_%C4%91o%C3%A1n), phát hiện [thẻ tín dụng giả](https://vi.wikipedia.org/w/index.php?title=Th%E1%BA%BB_t%C3%ADn_d%E1%BB%A5ng_gi%E1%BA%A3&action=edit&redlink=1), phân tích [thị trường chứng khoán](https://vi.wikipedia.org/wiki/Th%E1%BB%8B_tr%C6%B0%E1%BB%9Dng_ch%E1%BB%A9ng_kho%C3%A1n), phân loại các [chuỗi DNA](https://vi.wikipedia.org/w/index.php?title=Chu%E1%BB%97i_DNA&action=edit&redlink=1), [nhận dạng tiếng nói](https://vi.wikipedia.org/wiki/Nh%E1%BA%ADn_d%E1%BA%A1ng_ti%E1%BA%BFng_n%C3%B3i) và [chữ viết](https://vi.wikipedia.org/wiki/Nh%E1%BA%ADn_d%E1%BA%A1ng_ch%E1%BB%AF_vi%E1%BA%BFt_tay), [dịch tự động](https://vi.wikipedia.org/wiki/D%E1%BB%8Bch_t%E1%BB%B1_%C4%91%E1%BB%99ng), [chơi trò chơi](https://vi.wikipedia.org/wiki/Tr%C3%B2_ch%C6%A1i_chi%E1%BA%BFn_l%C6%B0%E1%BB%A3c) và [cử động rô-bốt](https://vi.wikipedia.org/w/index.php?title=C%E1%BB%AD_%C4%91%E1%BB%99ng_r%C3%B4-b%E1%BB%91t&action=edit&redlink=1) (robot locomotion).

**1.2.Phương pháp học máy có giám sát**

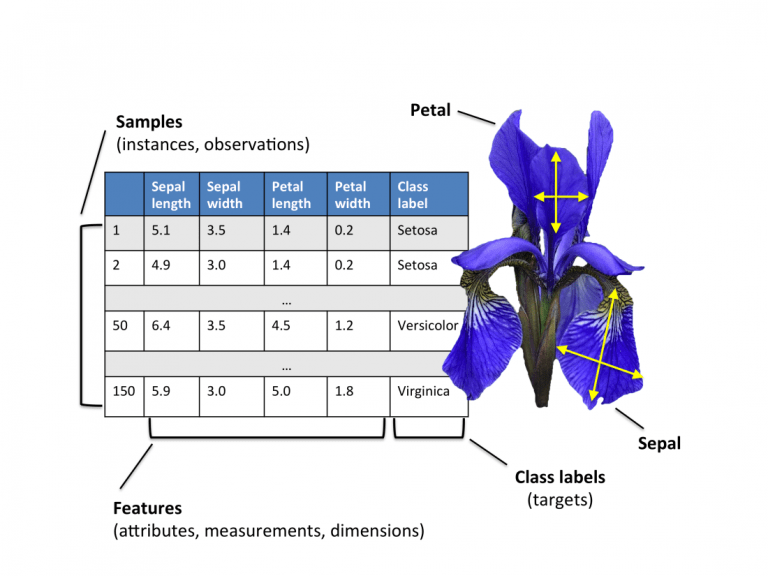
* + 1. **Học máy có giám sát là gì?**

Là một trong nhưng phương pháp học máy phổ biến nhất trong đó mô hình huấn luyện được thông qua mộ tập dữ liệu được gán nhãn trước đó.

Là một kỹ thuật của ngành học máy nhằm mục đích xây dựng một hàm f từ dữ tập dữ liệu huấn luyện (Training data). Dữ liệu huấn luyện bao gồm các cặp đối tượng đầu vào và đầu ra mong muốn. Đầu ra của hàm f có thể là một giá trị liên tục hoặc có thể là dự đoán một nhãn phân lớp cho một đối tượng đầu vào.

Nhiệm vụ của chương trình học có giám sát là dự đoán giá trị của hàm f cho một đối tượng đầu vào hợp lệ bất kì, sau khi đã xét một số mẫu dữ liệu huấn luyện (nghĩa là các cặp đầu vào và đầu ra tương ứng). Để đạt được điều này, chương trình học phải tổng quát hóa từ các dữ liệu sẵn có để dự đoán được những tình huống chưa gặp phải theo một cách hợp lý.

Chẳng hạn, [bộ dữ liệu hoa tử đằng (Iris)](https://archive.ics.uci.edu/ml/datasets/iris) chứa  các thuộc tính là chiều dài và chiều rộng của cánh hoa và đài hoa, các thuộc tính này tạo thành dữ liệu đầu vào (data). Đồng thời, nó cũng chứa cả nhãn (class label) của mục tiêu dự đoán (dòng hoa là một trong ba loại: setosa, versicolor và virginica).



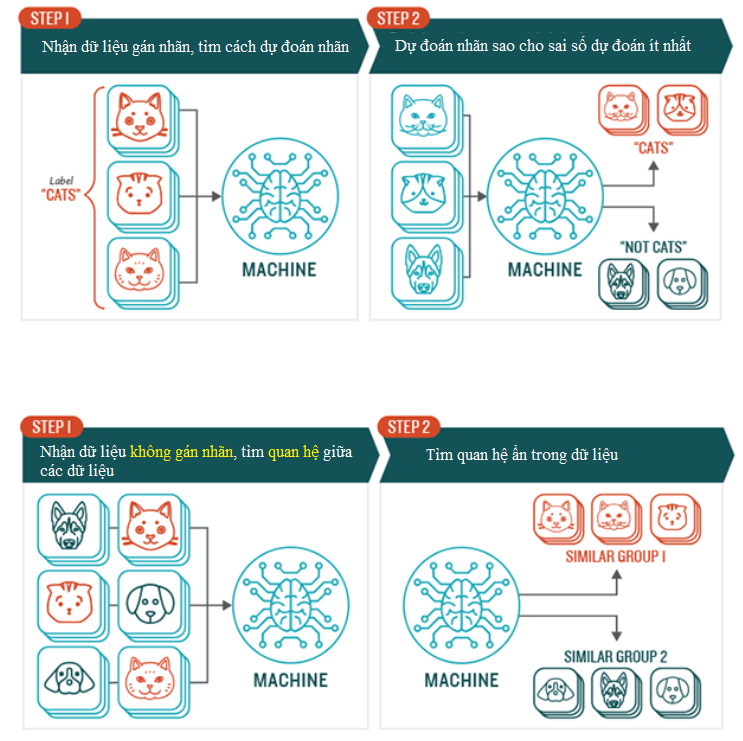
Bộ dữ liệu Iris, một trong những ví dụ kinh điển trong machine learning (Nguồn: Bishwamittra Ghosh)

Như ví dụ về mô hình hồi quy tuyến tính ở bài trước, bộ dữ liệu của chúng ta bao gồm cả mức lương và giá trị cần dự đoán là mức độ hài lòng với công việc của nhân viên. Trong các trường hợp này, một mô hình học có giám sát sẽ sử dụng bộ dữ liệu có bao gồm nhãn mục tiêu này để tìm cách dự đoán sao cho kết quả dự đoán đầu ra là chênh lệch ít nhất so với nhãn mục tiêu cho trước trong bộ dữ liệu. Quá trình này gọi là quá trình cực tiểu hóa sự sai khác giữa nhãn thật của dữ liệu và nhãn được dự đoán của mô hình đã xây dựng.

* + 1. **Mô hình và quy trình học có giám sát**
       1. **Mô hình học không giám sát (unsupervised learning)**

Học không giám sát là một lớp mô hình học sử dụng một thuật toán để mô tả hoặc trích xuất ra các mối quan hệ tiềm ẩn trong dữ liệu. Khác với học có giám sát, học không giám sát chỉ thực thi trên dữ liệu đầu vào không cần các thuộc tính nhãn, hoặc mục tiêu của việc học. Tức là không hề được cung cấp trước một kiến thức nào trước trừ dữ liệu. 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.

Hình dưới đây cho thấy sự khác biệt giữa học có giám sát và học không giám sát. Đối với học có giám sát chúng ta biết trước tên của các nhãn là "mèo" hoặc "không phải mèo" ttrong khi trong học không giám sát, tên của các nhãn không có trong bộ dữ liệu, chúng ta chỉ tìm được quan hệ rằng, có một số ảnh giống nhau ở phía trên (Similar Group 1) và một số ảnh giống nhau ở phía dưới (Simliar Group 2). Chúng ta không biết hai tập này là gì, và chỉ biết rằng chúng "giống nhau". Sau khi tìm được sự giống nhau này rồi, chúng ta vẫn có thể gán nhãn cho dữ liệu tương tự như bài toán phân lớp. Tuy nhiên, trong nhiều trường hợp, điều này đòi hỏi khá nhiều kiến thức của chuyên gia.



Có rất nhiều bài toán trong mô hình học không giám sát, nổi bật nhất là hai bài toán: phân cụm và xấp xỉ phân phối xác suất. Bài toán phân cụm có nhiệm vụ tìm kiếm các nhóm có tương quan trong dữ liệu (như ví dụ trên Hình trên), tức là có các đặc tính gần tương tự như nhau. Bài toán xấp xỉ phân phối tìm cách tổng hợp dạng phân phối của dữ liệu.

Một ví dụ về thuật toán phân cụm là k-Means, trong đó k đại diện cho số cụm cần tìm trong dữ liệu. Một ví dụ điển hình của các thuật toán xấp xỉ phân phối là thuật toán xấp xỉ mật độ nhân (Kernel Density Estimation) trong đó sử dụng một nhóm nhỏ mẫu dữ liệu có liên quan chặt chẽ với nhau để tìm cách xấp xỉ phân phối cho các điểm mới trong không gian dữ liệu của bài toán. Các mô hình này có thể được sử dụng để học ra các mẫu dạng trong dữ liệu mà không cần gán nhãn trước. Chẳng hạn các thư rác và thư thông thường có dạng khác nhau có thể phân thành các cụm khác nhau mà không cần biết trước nhãn của dữ liệu đầu vào.

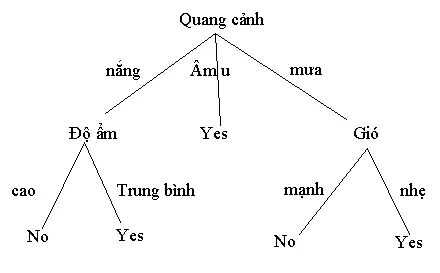
Ngoài ra, một số bài toán có thể được coi là học không giám sát chẳng hạn như bài toán biểu diễn dữ liệu (data visualization) hoặc các phương pháp chiếu dữ liệu (data projection). Biểu diễn dữ liệu là bài toán học không giám sát liên quan đến việc xây dựng các đồ thị và biểu đồ để biểu diễn trực quan dữ liệu. Còn các phương pháp chiếu cho phép tạo ra các biểu diễn dữ liệu có số chiều ít hơn dữ liệu gốc nhưng vẫn giữ được các đặc tính của dữ liệu gốc. Nó thường liên quan đến bài toán giảm chiều dữ liệu. Một đại diện thường gặp nhất của nó là phương pháp Phân tích thành phần chính (Principal Component Analysis) cho phép tổng hợp bộ dữ liệu thành các trị riêng và vector riêng trong đó loại bỏ các thành phần có quan hệ độc lập tuyến tính.

Nói tóm lại hai đặc điểm quan trọng nhất của unsupervised learning đó là dữ liệu không có dãn nhãn trước và mục tiêu để tìm ra các mối tương quan, các mẫu trong dữ liệu.

## Các thuật toán học có giám sát

## Cây quyết định (Decision Trees)

Cây quyết định là công cụ hỗ trợ quyết định sử dụng biểu đồ dạng cây hoặc mô hình của các quyết định và kết quả có thể xảy ra của chúng, bao gồm kết quả sự kiện ngẫu nhiên chi phí tài nguyên và lợi ích. Dưới đây là một ví dụ điển hình của cây quyết định:



Cây quyết định này cho ta gợi ý về việc có đi đá bóng hay không. Ví dụ, quang cảnh có nắng, độ ẩm trung bình thì tôi sẽ đi đá bóng. Ngược lại, nếu trời mưa, gió mạnh thì tôi sẽ không đi đá bóng nữa.

Cây quyết định tuy là mô hình khá cũ, khá đơn giản những vẫn còn được ứng dụng khá nhiều và hiệu quả. Đứng dưới góc nhìn thực tế, cây quyết định là một danh sách tối thiểu các câu hỏi dạng yes/no mà người ta phải hỏi, để đánh giá xác suất đưa ra quyết định đúng đắn.

* + - 1. **Phân loại Bayes (Naïve Bayes Classification)**

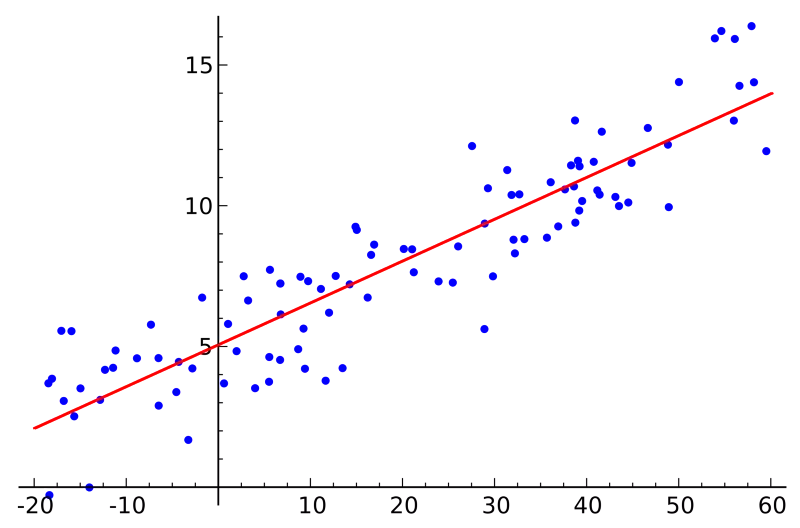
Phân loại Bayes là một nhóm các phân loại xác suất đơn giản dựa trên việc áp dụng định lý Bayes với các giả định độc lập (naïve) giữa các đặc tính.( bỏ ảnh hoặc tìm ảnh khác )

Trong đó: P(A|B) là xác suất có điều kiện A khi biết B, P(A) là xác suất giả thuyết A (tri thức có được về giải thuyết A trước khi có dữ liệu B), P(B|A) là xác suất có điều kiện B khi biết giả thuyết A, P(B) là xác suất của dữ liệu quan sát B không quan tâm đến bất kỳ giả thuyết A nào.

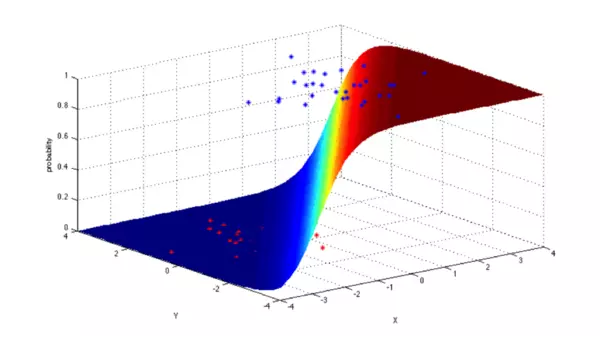
Thuật toán này được áp dụng trong một số bài toán như:

* Đánh dấu một email là spam hay không.
* Phân loại bài viết tin tức thuộc lĩnh vực công nghệ, chính trị hay thể thao.
* Kiểm tra một đoạn văn bản mang cảm xúc tích cực hay tiêu cực.
* Sử dụng cho các phần mềm nhận diện khuôn mặt. ...
  + - 1. **Hồi quy tuyến tính (Ordinary Least Squares Regression)**

1. Nếu bạn biết thống kê, bạn có thể đã nghe nói về hồi quy tuyến tính trước đây. Bình phương nhỏ nhất là một phương pháp để thực hiện hồi quy tuyến tính. Bạn có thể suy nghĩ về hồi quy tuyến tính như là nhiệm vụ kẻ một đường thẳng đi qua một tập các điểm. Có rất nhiều chiến lược có thể thực hiện được, và chiến lược "bình phương nhỏ nhất" sẽ như thế này - Bạn có thể vẽ một đường thẳng, và sau đó với mỗi điểm dữ liệu, đo khoảng cách thẳng đứng giữa điểm và đường thẳng. Đường phù hợp nhất sẽ là đường mà các khoảng cách này càng nhỏ càng tốt.



1. Một số ví dụ là người ta có thể sử dụng mô hình này để dự đoán giá cả (nhà đất, chứng khoán), điểm số,...
   * + 1. **Hồi quy logistic (Logistic Regression)**

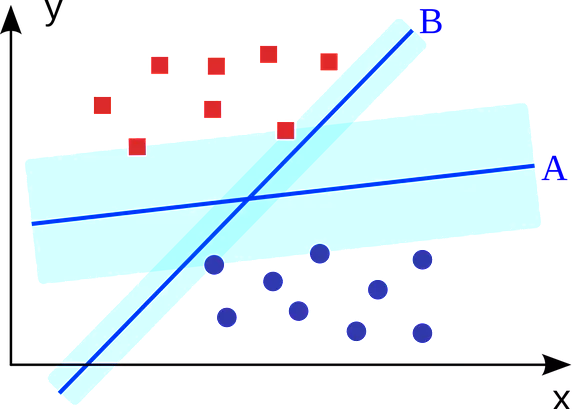
Hồi quy logistic là một cách thống kê mạnh mẽ để mô hình hóa một kết quả nhị thức với một hoặc nhiều biến giải thích. Nó đo lường mối quan hệ giữa biến phụ thuộc phân loại và một hoặc nhiều biến độc lập bằng cách ước tính xác suất sử dụng một hàm logistic, là sự phân bố tích lũy logistic.

Thuật toán này được sử dụng trong một số trường hợp:

* Điểm tín dụng ( quyết định có cho khách hàng vay vốn hay không)
* Đo mức độ thành công của chiến dịch marketing
* Dự đoán doanh thu của một sản phẩm nhất định
* Dự đoán động đất ....

### Support Vector Machines (SVM)

SVM là phương pháp phân loại nhị phân. Cho một tập các điểm thuộc 2 loại trong môi trường N chiều, SVM cố gắng tìm ra N-1 mặt phẳng để phân tách các điểm đó thành 2 nhóm. Ví dụ, cho một tập các điểm thuộc 2 loại như hình bên dưới, SVM sẽ tìm ra một đường thẳng nhằm phân cách các điểm đó thành 2 nhóm sao cho khoảng cách giữa đường thẳng và các điểm xa nhất có thể.

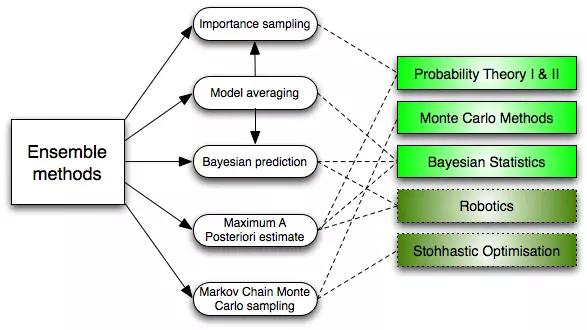


Xét về quy mô, một số vấn đề lớn nhất đã được giải quyết bằng cách sử dụng SVM (với việc thực hiện sửa đổi phù hợp) ví dụ như hiển thị quảng cáo, phát hiện giới tính dựa trên hình ảnh, phân loại hình ảnh có quy mô lớn ...

### Kết hợp các phương pháp (Ensemble Methods)

Phương pháp này dựa rên sự kết hợp của một vài phương pháp kể trên để dự đoán kết quả, sau đó sẽ đưa ra kết quả cuối cùng dựa vào trọng số của từng phương pháp

Vậy phương pháp này hoạt động như thế nào và tại sao nó lại ưu việt hơn các mô hình cá nhân?

Trung bình sai số (bias): một số phương pháp hoạt động tốt và cho sai số nhỏ, ngược lại cũng có một số phương pháp cho sai số lớn. Trung bình ta được một sai số chấp nhận được, có thể nhỏ hơn sai số khi sử dụng duy nhất một phương pháp.

* Giảm độ phụ thuộc vào tập dữ liệu (variance): ý kiến tổng hợp của một loạt các mô hình sẽ ít nhiễu hơn là ý kiến đơn lẻ của một mô hình. Trong lĩnh vực tài chính, đây được gọi là đa dạn hóa - một - một danh mục hỗn hợp của nhiều cổ phiếu sẽ ít biến động hơn so với chỉ một trong số các cổ phiếu riêng lẻ.
* Giảm over-fit: over-fit là hiện tượng khi mô hình hoạt động rất tốt với dữ liệu training, nhưng rất kém đối với dữ liệu test. Việc kết hợp nhiều mô hình cùng lúc giúp giảm vấn đề này.
  1. **Phương pháp học máy không có giám sát**

### 1.3.1.Học máy không có giám sát là gì?

Học máy không giám sát là nơi bạn chỉ có dữ liệu đầu vào (X) và không có biến đầu ra tương ứng.

Mục tiêu của việc học không giám sát là để mô hình hóa cấu trúc nền tảng hoặc sự phân bố trong dữ liệu để hiểu rõ hơn về nó.

Đây được gọi là học tập không giám sát vì không giống như việc học có giám sát ở trên, không có câu trả lời đúng và không có vị “giáo viên” nào cả. Các thuật toán được tạo ra chỉ để khám phá và thể hiện các cấu trúc hữu ích bên trong dữ liệu.

Các vấn đề học tập không giám sát có thể được phân ra thành hai việc chia nhóm và kết hợp.

Chia nhóm: Vấn đề về chia nhóm là nơi bạn muốn khám phá các nhóm vốn có bên trong dữ liệu, chẳng hạn như phân nhóm khách hàng theo hành vi mua hàng.

Kết hợp: Vấn đề về học tập quy tắc kết hợp là nơi bạn muốn khám phá các quy tắc mô tả dữ liệu của bạn, chẳng hạn như những người mua X cũng có khuynh hướng mua Y.

### 1.3.2. Mô hình và quy trình học máy không có giám sát

Mô hình học máy không có giám sát (unsupervised learning) là một phương pháp học máy mà không cần có dữ liệu được gắn nhãn. Trong học máy không có giám sát, mô hình phải tìm hiểu các mẫu trong dữ liệu mà không có bất kỳ thông tin giám sát nào từ bên ngoài. Phương pháp này thường được sử dụng để tìm ra cấu trúc ẩn trong dữ liệu, như phân cụm, giảm chiều dữ liệu, hoặc phát hiện ngoại lệ.

Có nhiều quy trình học máy không có giám sát được sử dụng, tùy thuộc vào mục đích của bài toán và loại dữ liệu được sử dụng. Dưới đây là một số quy trình phổ biến:

**1.Phân cụm (Clustering):** Quy trình này nhằm tìm kiếm các nhóm dữ liệu tương tự nhau trong tập dữ liệu. Kỹ thuật phân cụm phổ biến nhất là K-means.

**2.Giảm chiều dữ liệu (Dimensionality reduction):** Quy trình này giảm số chiều của dữ liệu trong khi vẫn giữ lại các đặc trưng quan trọng. Các kỹ thuật phổ biến nhất là PCA (Principal Component Analysis) và t-SNE (t-Distributed Stochastic Neighbor Embedding).

**3.Phát hiện ngoại lệ (Outlier detection):** Quy trình này nhằm xác định các điểm dữ liệu không giống với phần còn lại của tập dữ liệu. Các kỹ thuật phổ biến nhất là Local Outlier Factor (LOF) và Isolation Forest.

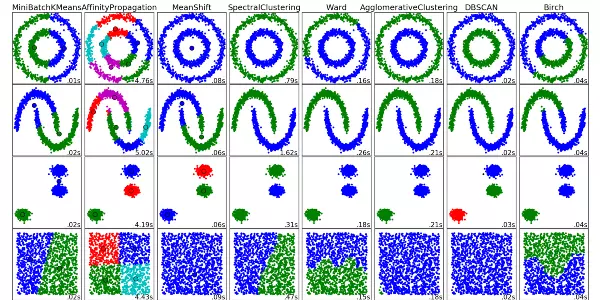
**4.Xác định kết hợp (Association rule learning):** Quy trình này tìm kiếm các mẫu kết hợp trong dữ liệu. Các kỹ thuật phổ biến nhất là Apriori và Eclat.

**5.Mô hình sinh dữ liệu (Generative models):** Quy trình này nhằm tạo ra các mẫu dữ liệu mới dựa trên phân phối xác suất của tập dữ liệu ban đầu. Các kỹ thuật phổ biến nhất là Variational Autoencoder và Generative Adversarial Networks (GANs).

### 1.3.3. Các thuật toán học không có giám sát

**1. Thuật toán gom cụm (Clustering Algorithms) (chỉ cần liệt kê )**

Gom cụm là nhiệm vụ nhóm một tập hợp các đối tượng sao cho các đối tượng trong cùng một nhóm (cluster) giống nhau hơn so với các đối tượng trong các nhóm khác.



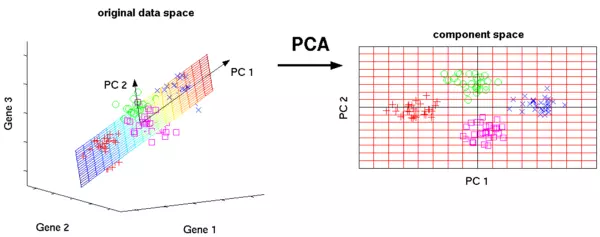
Gom cụm có nhiều phương pháp khác nhau, sau đây là một vài trong số đó:

* Gom cụm dựa vào tâm điểm (Centroid-based algorithms)
* Gom cụm dựa vào tính kết nối (Connectivity-based algorithms)
* Gom cụm dựa vào mật độ (Density-based algorithms)
* Gom cụm dựa vào xác suất (Probabilistic)
* Gom cụm dựa trên giảm chiều dữ liệu (Dimensionality Reduction)
* Gom cụm dựa trên mạng nơ-ron/deep leanring (Neural networks / Deep Learning)

**2. Phân tích thành phần chính (Principal Component Analysis - PCA)**

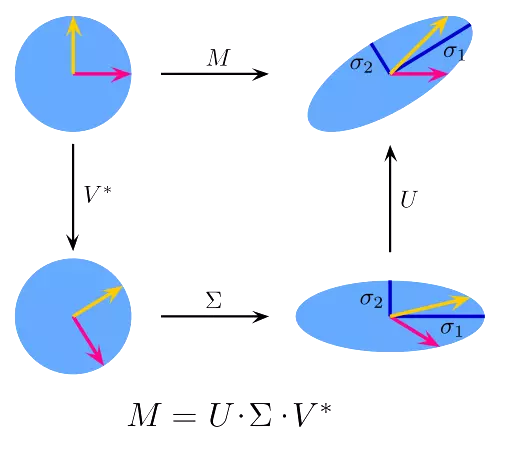
PCA là một thuật toán thống kê sử dụng phép biến đổi trực giao để biến đổi một tập hợp dữ liệu từ một không gian nhiều chiều sang một không gian mới ít chiều hơn (2 hoặc 3 chiều) nhằm tối ưu hóa việc thể hiện sự biến thiên của dữ liệu.

Phép biến đổi tạo ra những ưu điểm sau đối với dữ liệu:

* Giảm số chiều của không gian chứa dữ liệu khi nó có số chiều lớn, không thể thể hiện trong không gian 2 hay 3 chiều.
* Xây dựng những trục tọa độ mới, thay vì giữ lại các trục của không gian cũ, nhưng lại có khả năng biểu diễn dữ liệu tốt tương đương, và đảm bảo độ biến thiên của dữ liệu trên mỗi chiều mới.
* Tạo điều kiện để các liên kết tiềm ẩn của dữ liệu có thể được khám phá trong không gian mới, mà nếu đặt trong không gian cũ thì khó phát hiện vì những liên kết này không thể hiện rõ.
* Đảm bảo các trục tọa độ trong không gian mới luôn trực giao đôi một với nhau, mặc dù trong không gian ban đầu các trục có thể không trực giao.

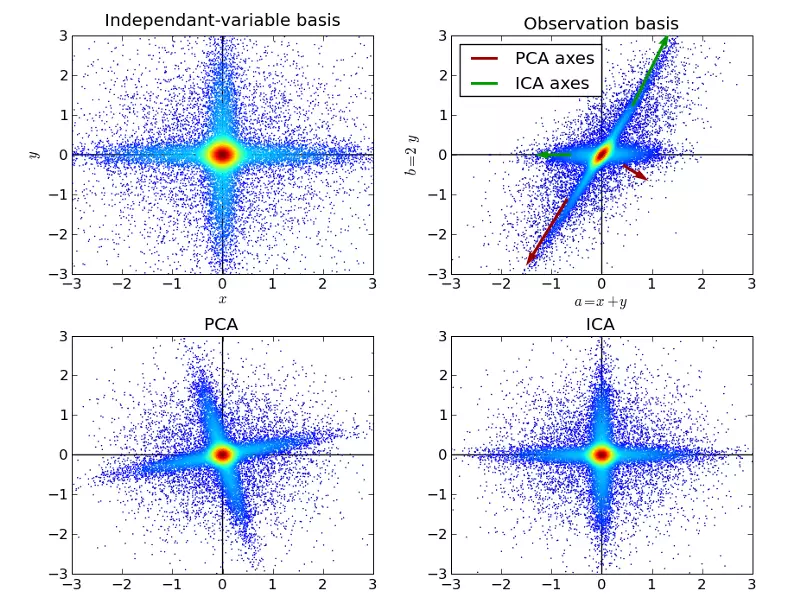
Một số ứng dụng của PCA bao gồm nén, đơn giản hóa dữ liệu để dễ dàng học tập, hình dung. Lưu ý rằng kiến thức miền là rất quan trọng trong khi lựa chọn có nên tiếp tục với PCA hay không. Nó không phù hợp trong trường hợp dữ liệu bị nhiễu (tất cả các thành phàn của PCA đều có độ biến thiên khá cao)

**3. Singular Value Decomposition**

Trong đại số tuyến tính, SVD là một thừa số của ma trận phức tạp thực sự. Đối với một ma trận m\*n đã xác định M, tồn tại một sự phân rã sao cho M = UΣV, trong đó U và V là các ma trận đơn nhất và Σ là một ma trận chéo.PCA thực ra là một ứng dụng đơn giản của SVD. Trong khoa học máy tính, các thuật toán nhận dạng khuôn mặt đầu tiên được sử dụng PCA và SVD để biểu diễn khuôn mặt như là một sự kết hợp tuyến tính của "eigenfaces", làm giảm kích thước, và sau đó kết hợp khuôn mặt với các tính chất thông qua các phương pháp đơn giản. Mặc dù các phương pháp hiện đại phức tạp hơn nhiều, nhiều người vẫn còn phụ thuộc vào các kỹ thuật tương tự.

**4. Phân tích thành phần độc lập (Independent Component Analysis)**

ICA là một kỹ thuật thống kê nhằm tìm ra các yếu tố ẩn nằm dưới các bộ biến ngẫu nhiên, các phép đo hoặc tín hiệu. ICA định nghĩa một mô hình phát sinh cho dữ liệu đa biến quan sát được, thường được đưa ra như một cơ sở dữ liệu lớn các mẫu. Trong mô hình, các biến số dữ liệu giả định là hỗn hợp tuyến tính của một số biến tiềm ẩn chưa biết, và hệ thống hỗn hợp cũng không rõ. Các biến tiềm ẩn được giả định không gaussian và độc lập với nhau, và chúng được gọi là các thành phần độc lập của dữ liệu được quan sát.



ICA có liên quan đến PCA, nhưng nó là một kỹ thuật mạnh hơn nhiều, có khả năng tìm ra các yếu tố bên dưới của các nguồn trong khi những phương pháp cổ điển thất bại hoàn toàn. Ứng dụng của nó bao gồm hình ảnh kỹ thuật số, cơ sở dữ liệu tài liệu, chỉ số kinh tế và đo lường tâm lý. Kết thúc bài viết ở đây, hi vọng bạn đọc đã có những cái nhìn tổng quan về các thuật toán phổ biến trong AI. Nếu cảm thấy thích thú, hãy đào sâu hơn về chúng để có thể tạo ra những ứng dụng có "trí tuệ nhân tạo" phục vụ cho mọi người.

# CHƯƠNG 2 . PHƯƠNG PHÁP RANDOM FOREST

* 1. **Random forest là gì?**

Random Forest là một tập hợp mô hình (ensemble). Mô hình Random Forest rất hiệu quả cho các bài toán phân loại vì nó huy động cùng lúc hàng trăm mô hình nhỏ hơn bên trong với quy luật khác nhau để đưa ra quyết định cuối cùng. Mỗi mô hình con có thể mạnh yếu khác nhau, nhưng theo nguyên tắc « wisdom of the crowd », ta sẽ có cơ hội phân loại chính xác hơn so với khi sử dụng bất kì một mô hình đơn lẻ nào.

**-Ví dụ:**

Lựa chọn chia ngẫu nhiên (Dietterich [1998]), tại mỗi một nút phép chia được lựa chọn ngẫu nhiên trong K phép chia tốt nhất. Breiman [1999] tạo một tập huấn luyện mới bằng cách lấy ngẫu nhiên đầu ra trong tập huấn luyện ban đầu. Một cách khác để lựa chọn tập huấn luyện từ một tập ngẫu nhiên của trọng số trên các ví dụ trong tập huấn luyện. Ho [1998] đã viết một số bài báo trên phương pháp " không gian con ngẫu nhiên - the random subspace" mà không lựa chọn ngẫu nhiên các tính năng của tập con sử dụng để phát triển mỗi cây.

Trong một bài báo quan trọng viết về nhận dạng ký tự, Amit and Geman [1997], xác định một số lượng lớn các tính năng hình học và tìm kiếm trong một lựa chọn ngẫu nhiên của chúng cho phân chia tốt nhất tại mỗi nút. Trong bài báo này đã ảnh hưởng suy nghĩ của tác giả.

Các phần tử phổ biến trong tất cả các quy trình là cho cây thứ k, một véc tơ ngẫu nhiên được tạo ra, độc lập với các véc tơ ngẫu nhiên trước đó , nhưng với phân bổ tương tự; và một cây sinh ra sử dụng tập huấn luyện và véc tơ ngẫu nhiên . Kết quả trong một lớpkhi *x* là một véc tơ đầu vào. Ví dụ, trong bagging một véc tơ ngẫu nhiên được tạo ra như đếm trong N hộp kết quả từ N lần lấy ngẫu nhiên tại các hộp khi N là số các ví dụ trong tệp huấn luyện. Lựa chọn chia ngẫu nhiên véc tơ gồm số các số nguyên ngẫu liên độc lập giữa 1 và K. Tính chất và chiều của véc tơ phụ thuộc vào sử dụng nó trong xây dựng cây.

Sau khi số lượng lớn các cây được tạo ra, chúng đề cử các lớp phổ biến nhất. Chúng ta gọi đó là quy trình **random forests.**

***Định nghĩa 1.1.*** *Một Random forest là một phân lớp gồm sự kết hợp cấu trúc cây phân lớp , khi là các véc tơ ngẫu nhiên phân bổ độc lập như nhau. Mỗi cây đại diện ứng viên cho lớp phổ biến nhiều nhất tại dữ liệu vào x.*

* **Đặc trưng độ chính xác của Random forest :**

Random Forests Converge (hội tụ)

Cho một quần thể của các phân lớp , và với tập huấn luyện được rút ra ngẫu nhiên từ sự phân bổ của véc tơ ngẫu nhiên X,Y, hàm xác định biên độ như là:



Với ** là hàm chỉ số. Biên độ đo mức độ số trung bình của các đề cử tại X,Y cho lớp đúng vượt quá số đề cử trung bình của bất kỳ lớp khác. Biên độ lớn, độ tin cậy lớn trong phân lớp. Lỗi được tổng hợp bởi công thức sau:



Với chỉ số dưới X,Y là xác suất trong không gian X,Y. Trong Random forest . Đối với số lượng lớn các cây sau từ Strong Law of Large Numbers và cấu trúc cây.

***Định lý 1.2.*** *Khi số lượng các cây tăng lên, gần như chắc chắn cho tất cả các dãy  hội tụ tới*

 (1)

***Chứng minh.*** Để thấy rằng một tập xác suất zero C trong không gian dãy **như vậy nằm ngoài C, cho tất cả các *x*



Cho một tập huấn luyện cố định và một véc tơ cố định, tập tất cả các *x* như  là sự kết hợp các hình chữ nhật siêu hình(hyper-rectangles). Cho tất cả các  chỉ giới hạn số K của sự kết hợp các hyper-rectangles, ký hiệu bởi *.* Định nghĩa . Nếu . Cho là số thời gian mà  trong N thử nghiệm đầu tiên. Thì



Bằng Law of Large Numbers,



Hội tụ tới . Lấy tất cả các tập mà trên đó không xẩy ra hội tụ đối với một số giá trị của k cho tập C có xác suất bằng 0. Do đó nằm bên ngoài C,



Giá trị đúng nằm vế phải là.

Kết quả này giải thích tại sao Random forest không overfit như các cây được thêm vào, nhưng tạo ra một giá trị giới hạn của lỗi tổng quát (generalization error).

* **Độ chắc chắn và Sự tương quan (Strength and Correlation) :**

Đối với Random forest cận trên có thể bắt nguồn cho các lỗi tổng quát trong hai tham số mà đó là biện pháp làm thế nào để xác thực các phân lớp độc lập và sự phụ thuộc giữa chúng. Sự tương tác giữa hai tham số cho cơ sở để hiểu hoạt động của Random forest. Chúng tôi dựa trên phân tích của Amit và Geman [1997].

***Định nghĩa 2.1.*** Hàm biên của Random forest là

 (2)

Và độ chắc chắn (*strength*) của tập các lớp  là

 (3)

Giả sử , bất đẳng thức của Chebyshev như sau:

 (4)

Một biểu thức thể hiện các phương sai của mr xuất phát như sau:

Cho 

Như vậy



***Định nghĩa 2.2.*** *The raw margin function is*



Với  là dự đoán của liên quan tới . Hàm f bất kỳ xác định



Với  độc lập với các phân bổ, ngụ ý rằng

 (5)

Sử dụng (5) cho

 (6)

Với là tương quan giữa ,  giữ cố định, và là độ lệch chuẩn của , giữ cố định.

Sau đó :

 (7)

Với  là giá trị trung bình tương quan, nghĩa là



Viết

 (8)

Đặt (4),(7),(8) như nhau.

Định lý 2.3. Giới hạn trên lỗi tổng quát được cho bởi



Mặc dù các ràng buộc có thể lỏng lẻo, nó tuân theo yêu cầu hàm của Random forest như giới hạn kiểuVC làm cho các kiểu khác của phân lớp. Nó cho thấy hai thành phần tham gia trong lỗi tổng hợp cho Random forest, và các mối tương quan giữa chúng trong điều kiện của hàm “raw margin”. Tỷ lệ là tưởng quan được phân chia bởi bình phương độ chắc chắn (strength). Trong hàm Random forest, cho biết tỷ lệ này nhỏ thì sẽ tốt hơn.

Định nghĩa 2.4. Tỷ lệ của hàm Random forest được xác định như sau:



Tỷ lệ này được đơn giản hóa trong hai tình huống. Hàm margin là:



Yêu cầu độ chắc chắn (strength) là tốt (xem (4)) trở thành tương tự như điều kiện huấn luyện yếu  Hàm margin tho là  và  là độ tương quan giữa  và . Thực tế, nếu các giá trị Y là +1 và -1 thì



Vì vậy  là độ tương quan giữa hai thành phần khác nhau của giá trị trung bình forest trên các phân bổ và 

Trong trường hợp lớn hơn hai lớp, phương pháp xác định độ tin cậy (strength) được xác định trong (3) phụ thuộc vào forest cũng như các cây độc lập vì các forest là xác định  Một cách viết khác :



Xác định



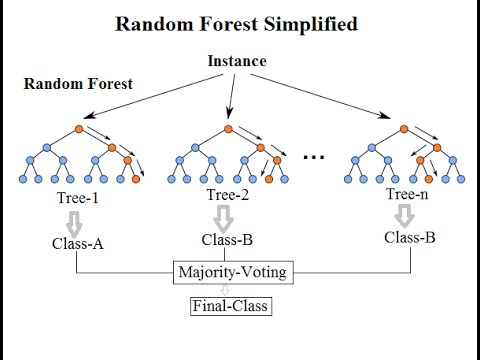
để có được độ tin cậy (strength) của các phân lớp  liên quan tới lớp *j*. Chú ý rằng xác định độ tin cậy (strength) không phụ thuộc trong forest. Sử dụng bất đẳng thức của Chebyshev, giả sử tất cả  dẫn đến

 (9)

và sử dụng xác định tương tự thông qua sử dụng xuất phápt từ (7), các biến trong (9) có thể được thể hiện trong các điều kiện tương quan trung bình. Ở đây không sử dụng các ước tính của (9) để nghiên cứu thực nghiệm, nhưng thú vị trong vấn đề đa lớp (multiple class).

* 1. **Mô hình và quy trình học phương pháp Random forest**
     1. **Mô hình Random Forest:**

- Mô hình Random Forest bao gồm một tập hợp các cây quyết định đơn lẻ. Khi tạo một cây quyết định, tập dữ liệu được chia thành các tập con ngẫu nhiên, mỗi tập con được sử dụng để xây dựng một cây quyết định. Các cây quyết định này sẽ có sự khác nhau nhẹ trong cách mà chúng được xây dựng và được sử dụng để đưa ra dự đoán. Khi có một dữ liệu mới cần được phân loại hoặc dự đoán, các cây quyết định trong mô hình Random Forest sẽ đưa ra dự đoán và kết quả cuối cùng được tính toán bằng cách sử dụng đa số phiếu bầu từ các cây quyết định.



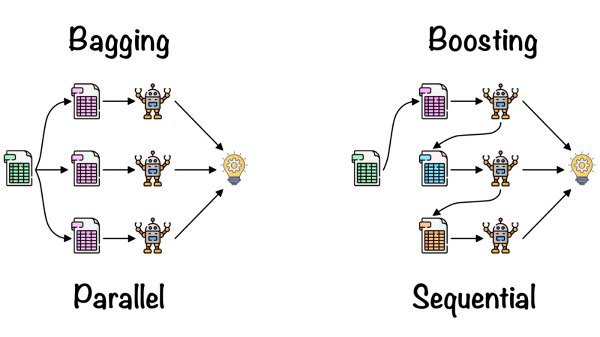
Hoạt động của thuật toán rừng ngẫu nhiên

Trước khi hiểu hoạt động của thuật toán rừng ngẫu nhiên trong học máy, chúng ta phải xem xét kỹ thuật học tập đồng bộ. Ensemble đơn giản có nghĩa là kết hợp nhiều mô hình. Do đó, một tập hợp các mô hình được sử dụng để đưa ra dự đoán thay vì một mô hình riêng lẻ.

Ensemble sử dụng hai loại phương pháp:

**1. Bagging** nó tạo ra một tập hợp con đào tạo khác với dữ liệu đào tạo mẫu có thay thế & kết quả cuối cùng dựa trên biểu quyết đa số. Ví dụ: Rừng ngẫu nhiên.

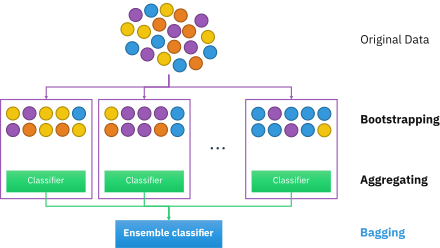
**2. Boosting nó** kết hợp những người học yếu thành những người học mạnh bằng cách tạo các mô hình tuần tự sao cho mô hình cuối cùng có độ chính xác cao nhất. Ví dụ: ADA BOOST, XG BOOST.



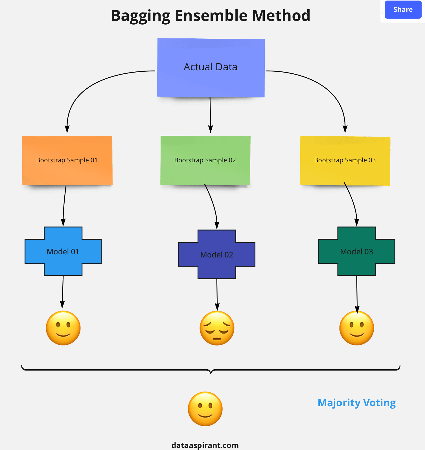
Như đã đề cập trước đó, Rừng ngẫu nhiên hoạt động theo nguyên tắc Đóng bao. Bây giờ, hãy đi sâu vào và hiểu chi tiết về việc đóng bao.

**BAGGING**

Bagging , còn được gọi là Tập hợp Bootstrap, là kỹ thuật tập hợp được rừng ngẫu nhiên sử dụng. Đóng bao chọn một mẫu ngẫu nhiên/tập hợp con ngẫu nhiên từ toàn bộ tập dữ liệu. Do đó, mỗi mô hình được tạo từ các mẫu (Mẫu Bootstrap) do Dữ liệu gốc cung cấp với sự thay thế được gọi là lấy mẫu hàng . Bước lấy mẫu hàng có thay thế này được gọi là bootstrap . Bây giờ mỗi mô hình được đào tạo độc lập, tạo ra kết quả. Kết quả cuối cùng dựa trên biểu quyết đa số sau khi kết hợp kết quả của tất cả các mô hình. Bước này liên quan đến việc kết hợp tất cả các kết quả và tạo đầu ra dựa trên biểu quyết đa số, được gọi là tổng hợp .



Bây giờ hãy xem một ví dụ bằng cách chia nhỏ nó ra với sự trợ giúp của hình sau. Ở đây, mẫu bootstrap được lấy từ dữ liệu thực tế (mẫu Bootstrap 01, mẫu Bootstrap 02 và mẫu Bootstrap 03) có thay thế, nghĩa là có khả năng cao là mỗi mẫu sẽ không chứa dữ liệu duy nhất. Mô hình (Mô hình 01, Mô hình 02 và Mô hình 03) thu được từ mẫu bootstrap này được đào tạo độc lập. Mỗi mô hình tạo ra kết quả như được hiển thị. Giờ đây, biểu tượng cảm xúc Vui vẻ chiếm đa số khi so sánh với biểu tượng cảm xúc Buồn. Do đó, dựa trên đa số biểu quyết, kết quả cuối cùng thu được dưới dạng biểu tượng cảm xúc Hạnh phúc.



**BOOSTING**

Boosting là một trong những kỹ thuật sử dụng khái niệm học tập đồng bộ. Thuật toán tăng cường kết hợp nhiều mô hình đơn giản (còn được gọi là trình học yếu hoặc công cụ ước tính cơ sở) để tạo ra kết quả cuối cùng. Nó được thực hiện bằng cách xây dựng một mô hình bằng cách sử dụng các mô hình yếu theo chuỗi.

Có một số thuật toán tăng cường; AdaBoost là thuật toán tăng cường thực sự thành công đầu tiên được phát triển cho mục đích phân loại nhị phân. AdaBoost là tên viết tắt của Adaptive Boosting và là một kỹ thuật tăng cường phổ biến kết hợp nhiều “bộ phân loại yếu” thành một “bộ phân loại mạnh” duy nhất. Có các kỹ thuật Tăng cường khác.

### 2.2.2. Quy trình học random forest

**1. Chuẩn bị dữ liệu:** Thu thập và xử lý dữ liệu không chính xác, không đầy đủ và dũa liệu méo mó; chia dữ liệu thành tập hai tập: Một tập gốm 70% dữ liệu sử dụng huấn luyện mô hình và tập còn lại là dữ liệu sử để kiểm tra mô hình.

**2. Xây dựng cây** bằng phương pháp CART.

**3. Lựa chọn cây :** Số lượng cây trong mô hình Random Forest có thể được lựa chọn bằng cách sử dụng các kỹ thuật cross-validation để tìm ra số lượng cây tối ưu trong mô hình.

**4. Huấn luyện mô hình Random Forest:** Mô hình Random Forest được huấn luyện bằng cách lặp lại quá trình xây dựng cây quyết định đơn lẻ và tính hợp các cây quyết định độc lập lại vơi snhau và có tính đa dạng , giúp giảm thiểu sự quá khớp( overfitting) và khả năng tang dự đoán của mô hình

**5. Đánh giá mô hình Random forest :** Sau khi huấn luyện xong, mô hình Random Forest được đánh giá bằng cash sử dụng tập kiểm tra. Kết quả của mô hình được so sánh với các giá trị thực tế để đánh giá độ chính xác và hiệu suất của mô hình.

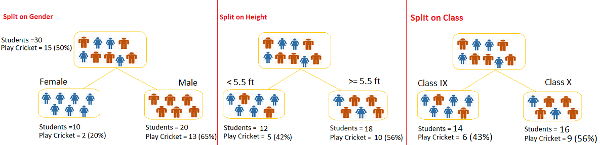
**6. Tinh chỉnh mô hình** : Nếu mô hình Random Forest không đạt được kết quả tốt , chúng ta có thể tinh chỉnh mô hình bằng cách thay dổi các siêu tham số của thuật toán , chẳng hạn như số lượng cây trong mô hình, độ sâu của cây quyết định và các tham số khác .

**7. Dự đoán** : Sau khi mô hình Random forest được huấn luyện và tinh chỉnh , chúng ta có thể sử dụng nó để phân loại hoặc có dự đoán bằng cách đưa bộ dữ liệu mới..

**Ví dụ:-**

Giả sử chúng ta có một mẫu gồm 30 học sinh với ba biến Giới tính (Nam/Nữ), Lớp (IX/X) và Chiều cao (5 đến 6 ft). 15 trong số 30 người này chơi cricket trong thời gian rảnh rỗi. Bây giờ, tôi muốn tạo một mô hình để dự đoán ai sẽ chơi cricket trong thời gian rảnh rỗi? Trong vấn đề này, chúng ta cần tách biệt những sinh viên chơi cricket trong thời gian rảnh rỗi dựa trên biến đầu vào có ý nghĩa cao giữa cả ba.

Đây là nơi cây quyết định giúp ích, nó sẽ phân tách các sinh viên dựa trên tất cả các giá trị của ba biến và xác định biến, tạo ra các tập hợp sinh viên đồng nhất tốt nhất (không đồng nhất với nhau). Trong ảnh chụp nhanh bên dưới, bạn có thể thấy rằng biến Giới tính có thể xác định các tập hợp đồng nhất tốt nhất so với hai biến còn lại.

[](https://www.analyticsvidhya.com/wp-content/uploads/2015/01/Test.png)

Như đã đề cập ở trên, cây quyết định xác định biến quan trọng nhất và giá trị của nó mang lại các tập hợp dân số đồng nhất tốt nhất. Bây giờ câu hỏi đặt ra là, làm thế nào để nó xác định biến và sự phân chia? Để làm được điều này, cây quyết định sử dụng nhiều thuật toán khác nhau mà chúng ta sẽ thảo luận trong phần sau.

## Tại sao lại sử dụng Random Forest :

Tại sao lại sử dụng Random Forest Random forest là một phương pháp học máy phổ biến trong việc xây dựng các mô hình dự đoán. Đây là một phương pháp ensemble learning, tức là nó kết hợp nhiều mô hình dự đoán đơn lẻ để tạo ra một mô hình dự đoán mạnh mẽ hơn.

Các lợi ích của việc sử dụng random forest bao gồm:

**1.** Khả năng xử lý các tập dữ liệu lớn: Random forest có thể xử lý các tập dữ liệu lớn mà không làm giảm hiệu suất của mô hình.

**2.** Khả năng dự đoán chính xác: Random forest cung cấp các dự đoán chính xác, đặc biệt là trong trường hợp của các bài toán phân loại và dự đoán.

**3.** Khả năng xác định mức độ quan trọng của các tính năng: Random forest có thể giúp xác định các tính năng quan trọng nhất trong các tập dữ liệu. Điều này rất hữu ích để hiểu rõ hơn về cách các tính năng ảnh hưởng đến các kết quả dự đoán.

**4.** Khả năng giảm thiểu sự quá khớp (overfitting): Random forest sử dụng các mô hình dự đoán đơn lẻ để giảm thiểu sự quá khớp trong mô hình dự đoán. Tóm lại, random forest là một phương pháp học máy mạnh mẽ và đa năng, có thể được sử dụng để giải quyết nhiều bài toán dự đoán khác nhau, đặc biệt là trong trường hợp các tập dữ liệu lớn và phức tạp. Tiếp Ngoài những lợi ích đã nêu ở trên, random forest còn có những đặc tính và ưu điểm khác như sau:

**5.** Độ tin cậy cao: Vì random forest là sự kết hợp của nhiều mô hình dự đoán độc lập, nên nó đưa ra các dự đoán với độ tin cậy cao hơn so với một mô hình đơn lẻ.

**6.** Khả năng xử lý các tính năng đầu vào thiếu: Random forest có khả năng xử lý các tập dữ liệu có tính năng đầu vào thiếu một cách hiệu quả. Dễ dàng sử dụng và triển khai: Random forest là một phương pháp học máy phổ biến, được cung cấp trong các thư viện phân tích dữ liệu phổ biến như Scikit-learn của Python, và có thể được triển khai dễ dàng trong nhiều ứng dụng thực tế.

Tuy nhiên, cũng có những hạn chế khi sử dụng random forest, bao gồm:

**7.** Khả năng mô hình bị overfitting: Random forest có khả năng bị overfitting nếu số cây trong forest quá lớn hoặc số lượng tính năng đầu vào quá ít.

**8.** Khả năng mô hình bị chậm: Với những tập dữ liệu lớn, quá trình huấn luyện và dự đoán với random forest có thể trở nên chậm hơn so với một số phương pháp khác.

**=>** Tóm lại, random forest là một phương pháp học máy mạnh mẽ và đa năng, có thể giải quyết nhiều bài toán dự đoán khác nhau, đặc biệt là trong trường hợp các tập dữ liệu lớn và phức tạp. Tuy nhiên, để tận dụng hết các ưu điểm của phương pháp này, cần phải lựa chọn số lượng cây và tính năng đầu vào phù hợp.

**Bình luận và kết luận:**

Random forest là một công cụ hiệu quả trong dự đoán bởi vì luật số lớn (Law of Large Numbers) không phù hợp. Injecting đúng loại ngẫu nhiên tạo ra tính xác thực của classification và regression. Hơn nữa trong khuôn khổ về độ chắc chắn (strength) của các dự đoán độc lập và mỗi tương quan (correclation) mang lại cái nhìn sâu sắc vào khả năng dự báo random forest. Sử dụng ước tính Out-of-bag làm nền tảng cho giá trị lý thuyết khác nhau của độ chắc chắn (strength) và sự tương quan (correclation)

Trong một thời gian, những suy nghĩ thông thường là forest không thể so sánh về độ chính xác so với các thuật toán kiểu arcing. Kết quả của chúng tôi loại bỏ cái định kiến này, nhưng dẫn đến một câu hỏi thú vị. Boosting và các thuật toán arcing( hồ quang) có khả năng làm giản độ lệch (bias) cũng như phương sai (variance) (Schapire et al [1998]). Thuật toán “adaptive bagging” trong regression (Breiman [1999]) được thiết kế để làm giảm độ lệch (bias) và hoạt động hiệu quả trong classification cũng như trong Regression.

Nhưng giống arcing, nó cũng thay đổi tập huấn luyện khi xử lý nó.

Forset cho kết quả cạnh tranh với boosting và “adaptive bagging” nhưng từng bước thay đổi tập huấn luyện. Độ chính xác cho thấy rằng độ sai lệch (bias) giảm. Các cơ chế này không rõ ràng. Random forest cũng có thể được xem như thủ tục Bayer, mặc dù tôi nghi ngờ rằng đây là một thăm dò hiệu quả, nếu có thể giả thích độ lệch (bias) giảm, nó có thể trở thành một số Bayer.

Các đầu vào ngẫu nhiên và các tính năng ngẫu nhiên tạo ra một kết quả tốt trong phân lớp, kém hơn trong hồi quy. Kiểu duy nhất ngẫu nhiên được sử dụng trong trường hợp nghiên cứu này là bagging và tính năng ngẫu nhiên. Nó có thể có các kiểu khác nhau nhẫu nhiên cho kết quả tốt hơn. Ví dụ một trong những trọng tài đề nghị tính năng kết hợp Boolean ngẫu nhiên .

Một câu hỏi gần như rỏ ràng liệu tăng độ chính xác có thể kết hợp bởi tính năng ngẫu nhiên và boosting. Cho một tập dữ liệu lớn, một tỷ lệ lỗi thấp là có thể. Trên một vài bước chạy, chúng tôi nhận được lỗi thấp là 5.1% trên dữ liệu zip-code, 2.2% trên dữ liệu letter và 7.9% trên dữ liệu satellite, sự cải thiện ít trên tập dữ liệu smaller, nhiều công việc cần điều này nhưng nó cho thấy rằng các việc injections khác nhau ngẫu nhiên có thể tạo ra kết quả tốt hơn.

# CHƯƠNG 3 . BÀI TOÁN ỨNG DỤNG

* 1. **Mô tả dữ liệu (Mô tả một bộ dữ liệu thử nghiệm trên UCI)**

UCI Machine Learning Repository là một trang web chứa các bộ dữ liệu thử nghiệm được sử dụng phổ biến trong cộng đồng máy học. Dưới đây là một mô tả về một bộ dữ liệu thử nghiệm trên UCI:

Tên: Iris Data Set

Mô tả: Bộ dữ liệu Iris là bộ dữ liệu được sử dụng phổ biến trong Machine Learning. Bộ dữ liệu này bao gồm 3 loại hoa iris (Setosa, Versicolour và Virginica), mỗi loại gồm 50 mẫu. Với mỗi mẫu, các đặc trưng được đo lường bao gồm: chiều dài cánh hoa, chiều rộng cánh hoa, chiều dài đài hoa và chiều rộng đài hoa, được đo bằng đơn vị cm.

Số lượng mẫu: 150

Số lượng đặc trưng: 4

File dữ liệu: iris.data

Đây là một bộ dữ liệu cơ bản được sử dụng phổ biến để giới thiệu các thuật toán phân loại. Nhiều thuật toán phân loại đã được áp dụng trên bộ dữ liệu này và đưa ra các kết quả tương đối tốt. Bộ dữ liệu này rất hữu ích cho việc học tập và thực hành trong lĩnh vực Machine Learning.

* 1. **Xử lí dữ liệu học**

Xử lí dữ liệu học (hay còn gọi là data preprocessing) là quá trình chuẩn bị và xử lí dữ liệu để chúng có thể được sử dụng hiệu quả trong các mô hình học máy và trí tuệ nhân tạo. Các bước xử lí dữ liệu học thường bao gồm:

* Thu thập dữ liệu: Bước đầu tiên là thu thập dữ liệu từ các nguồn khác nhau, bao gồm cả các bộ dữ liệu công khai và dữ liệu do người dùng cung cấp.
* Làm sạch dữ liệu: Sau khi thu thập dữ liệu, chúng ta cần kiểm tra và loại bỏ các giá trị bị thiếu, giá trị nhiễu, hoặc giá trị không hợp lệ.
* Tiền xử lí dữ liệu: Tiền xử lí dữ liệu là bước để biến đổi dữ liệu thành định dạng phù hợp cho mô hình học máy. Bao gồm các bước như chuẩn hóa, mã hóa và chuyển đổi đặc trưng của dữ liệu.
* Phân chia tập huấn luyện và tập kiểm tra: Dữ liệu cần được chia thành hai phần, phần dùng để huấn luyện mô hình và phần dùng để đánh giá mô hình.
* Thử nghiệm mô hình và điều chỉnh tham số: Sau khi mô hình được huấn luyện, chúng ta cần thử nghiệm và đánh giá hiệu quả của nó. Nếu cần thiết, chúng ta có thể điều chỉnh tham số của mô hình để tối ưu hóa hiệu suất.
* Xử lí mất cân bằng dữ liệu: Một số bộ dữ liệu có thể bị mất cân bằng giữa các lớp, điều này có thể ảnh hưởng đến hiệu suất của mô hình học máy. Vì vậy, ta có thể sử dụng các phương pháp như oversampling, undersampling hoặc smote để xử lí mất cân bằng dữ liệu.
* Xử lí dữ liệu bị nhiễu: Dữ liệu có thể bị nhiễu, chứa giá trị ngoại lai hoặc nhiễu, làm giảm độ chính xác của mô hình học máy. Để xử lý điều này, ta có thể sử dụng các phương pháp như xóa bỏ nhiễu, lọc nhiễu hoặc sử dụng mô hình học máy khác để phát hiện và loại bỏ nhiễu.
* Sử dụng phương pháp k-fold cross-validation: Phương pháp k-fold cross-validation là một kỹ thuật đánh giá hiệu suất của mô hình học máy. Nó chia dữ liệu thành k phần, sử dụng k-1 phần để huấn luyện và phần còn lại để đánh giá hiệu suất của mô hình.
* Sử dụng các kỹ thuật giảm chiều dữ liệu: Các kỹ thuật giảm chiều dữ liệu giúp giảm số lượng đặc trưng của dữ liệu, làm cho quá trình huấn luyện nhanh hơn và giảm thiểu overfitting. Các kỹ thuật này bao gồm PCA, LDA và t-SNE.
* Sử dụng mô hình học máy phù hợp: Cuối cùng, khi đã hoàn thành các bước xử lý dữ liệu học, ta cần chọn mô hình học máy phù hợp để huấn luyện dữ liệu. Các mô hình phổ biến bao gồm hồi quy tuyến tính, phân loại, cây quyết định, SVM, mạng nơ-ron và học sâu.

. => Việc xử lí dữ liệu học là một bước quan trọng để đảm bảo rằng mô hình học máy hoạt động tốt trên dữ liệu mới và cung cấp kết quả chính xác.

* 1. **Thử nghiệm và đánh giá mô hình học**

## Triển khai hồi quy rừng ngẫu nhiên trong Python

Mục tiêu của chúng tôi ở đây là xây dựng một nhóm cây quyết định, mỗi cây đưa ra dự đoán về biến phụ thuộc và dự đoán cuối cùng của rừng ngẫu nhiên là trung bình dự đoán của tất cả các cây.

Trong ví dụ của chúng tôi, chúng tôi sẽ sử dụng bộ dữ liệu Lương - vị trí sẽ dự đoán mức lương dựa trên dự đoán.

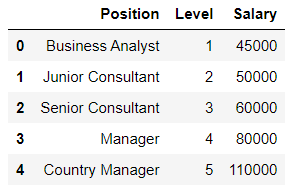
Tập dữ liệu được sử dụng có thể được tìm thấy tại <https://github.com/content-anu/dataset-polynomial-regression>

**1. Nhập tập dữ liệu**

Chúng ta sẽ sử dụng các thư viện [numpy](https://www.askpython.com/python-modules/numpy/python-numpy-module), [pandas](https://www.askpython.com/python-modules/pandas/python-pandas-module-tutorial) và [matplotlib](https://www.askpython.com/python-modules/matplotlib/python-matplotlib) để triển khai mô hình của chúng ta.

|  |
| --- |
| import pandas as pd  import numpy as np  import matplotlib.pyplot as plt    dataset = pd.read\_csv('Position\_Salaries.csv')  dataset.head() |

Ảnh chụp nhanh tập dữ liệu như sau:

Ảnh chụp nhanh đầu ra của tập dữ liệu

**2. Tiền xử lý dữ liệu**

Chúng tôi sẽ không có nhiều tiền xử lý dữ liệu. Chúng ta sẽ chỉ phải xác định ma trận các tính năng và mảng vectơ hóa.

|  |
| --- |
| X = dataset.iloc[:,1:2].values  y = dataset.iloc[:,2].values |

**3. Điều chỉnh hồi quy rừng ngẫu nhiên vào tập dữ liệu**

Chúng tôi sẽ nhập RandomForestRegressor từ thư viện tổng hợp của sklearn. Chúng ta tạo một đối tượng hồi quy bằng cách sử dụng hàm tạo lớp RFR. Các thông số bao gồm:

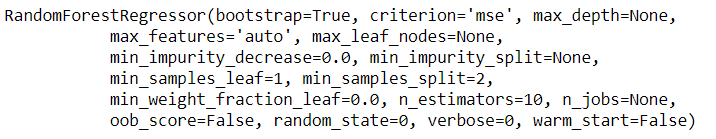
n\_estimators : số lượng cây trong rừng. (mặc định = 10)

Tiêu chí: Mặc định là mse tức là lỗi bình phương trung bình. Đây cũng là một phần của cây quyết định.

random\_state

|  |
| --- |
| from sklearn.ensemble import RandomForestRegressor  regressor = RandomForestRegressor(n\_estimators = 10, random\_state = 0)  regressor.fit(X,y) |

**Đường hồi quy như sau:**



Đường hồi quy

Chúng tôi sẽ chỉ đưa ra dự đoán thử nghiệm như sau:

y\_pred=regressor.predict([[6.5]])

y\_pred

Hình ảnh 71Đầu ra của dự đoán

**4. Hình dung kết quả**

#higher resolution graph

X\_grid = np.arange(min(X),max(X),0.01)

X\_grid = X\_grid.reshape(len(X\_grid),1)

plt.scatter(X,y, color='red') #plotting real points

plt.plot(X\_grid, regressor.predict(X\_grid),color='blue') #plotting for predict points

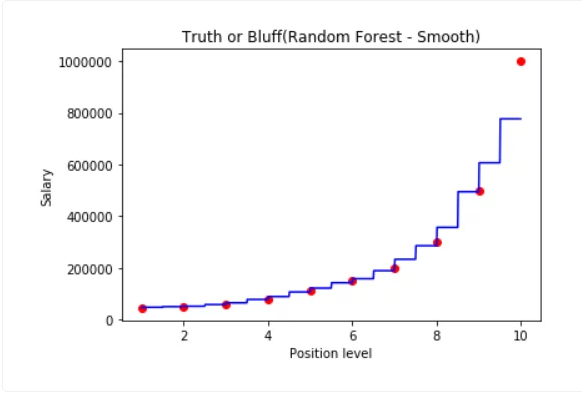
plt.title("Truth or Bluff(Random Forest - Smooth)")

plt.xlabel('Position level')

plt.ylabel('Salary')

plt.show()

Biểu đồ được tạo ra như hình dưới đây:



Biểu đồ cho 10 cây

**5. Giải thích biểu đồ trên**

Chúng tôi nhận được nhiều bước trong biểu đồ này hơn là với một cây quyết định. Chúng tôi có nhiều khoảng thời gian và phân chia hơn. Chúng tôi nhận được nhiều bước hơn trong cầu thang của chúng tôi.

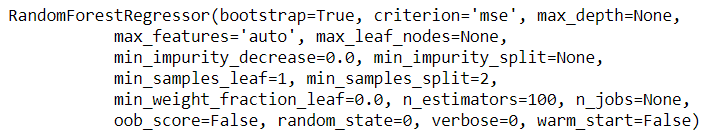
Mỗi dự đoán dựa trên 10 phiếu bầu (chúng tôi đã thực hiện 10 cây quyết định). Rừng ngẫu nhiên tính toán nhiều mức trung bình cho mỗi khoảng thời gian này.

Chúng tôi càng bao gồm nhiều cây, độ chính xác càng nhiều vì nhiều cây hội tụ đến cùng một mức trung bình cuối cùng.

**6. Xây dựng lại mô hình cho 100 cây xanh**

|  |
| --- |
| from sklearn.ensemble import RandomForestRegressor  regressor = RandomForestRegressor(n\_estimators = 100, random\_state = 0)  regressor.fit(X,y) |

Phương trình hồi quy được hình thành cho 100 cây trên như sau:

Phương trình hồi quy

**7. Tạo biểu đồ cho 100 cây**

|  |
| --- |
| #higher resolution graph  X\_grid = np.arange(min(X),max(X),0.01)  X\_grid = X\_grid.reshape(len(X\_grid),1)  plt.scatter(X,y, color='red')    plt.plot(X\_grid, regressor.predict(X\_grid),color='blue')  plt.title("Truth or Bluff(Random Forest - Smooth)")  plt.xlabel('Position level')  plt.ylabel('Salary')  plt.show() |
|  |

Biểu đồ với 100 cây

Các bước của biểu đồ không tăng gấp 10 lần số cây trong rừng. Nhưng dự đoán sẽ tốt hơn. Hãy dự đoán kết quả của cùng một biến.

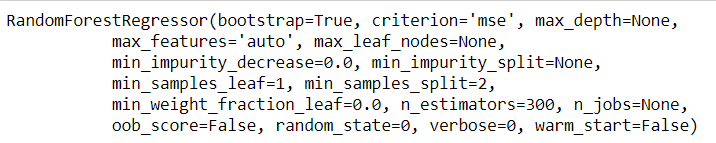
|  |
| --- |
| y\_pred=regressor.predict([[6.5]])  y\_pred |

Hình ảnh 75Dự đoán đầu ra

**8. Xây dựng lại mô hình cho 300 cây xanh**

|  |
| --- |
| from sklearn.ensemble import RandomForestRegressor  regressor = RandomForestRegressor(n\_estimators = 300, random\_state = 0)  regressor.fit(X,y) |

Đầu ra cho đoạn mã trên tạo ra hồi quy sau:



Hồi quy cho 300 cây

**9. Biểu đồ cho 300 cây**

#higher resolution graph

X\_grid = np.arange(min(X),max(X),0.01)

X\_grid = X\_grid.reshape(len(X\_grid),1)

plt.scatter(X,y, color='red') #plotting real points

plt.plot(X\_grid, regressor.predict(X\_grid),color='blue') #plotting for predict points

plt.title("Truth or Bluff(Random Forest - Smooth)")

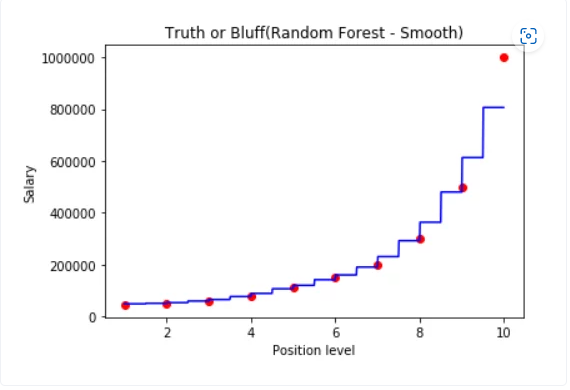
plt.xlabel('Position level')

plt.ylabel('Salary')

plt.show()

|  |
| --- |
|  |

Đoạn mã trên tạo ra biểu đồ sau:



Biểu đồ cho 300 cây

Bây giờ, chúng ta hãy đưa ra một dự đoán.

|  |
| --- |
| y\_pred=regressor.predict([[6.5]])  y\_pred |

Đầu ra cho đoạn mã trên như sau:

Hình ảnh 78Dự đoán sử dụng 300 cây

**Hoàn thành mã Python để thực hiện hồi quy rừng ngẫu nhiên**

|  |
| --- |
|  |

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

dataset = pd.read\_csv('Position\_Salaries.csv')

dataset.head()

X = dataset.iloc[:,1:2].values

y = dataset.iloc[:,2].values

# for 10 trees

from sklearn.ensemble import RandomForestRegressor

regressor = RandomForestRegressor(n\_estimators = 10, random\_state = 0)

regressor.fit(X,y)

y\_pred=regressor.predict([[6.5]])

y\_pred

#higher resolution graph

X\_grid = np.arange(min(X),max(X),0.01)

X\_grid = X\_grid.reshape(len(X\_grid),1)

plt.scatter(X,y, color='red') #plotting real points

plt.plot(X\_grid, regressor.predict(X\_grid),color='blue') #plotting for predict points

plt.title("Truth or Bluff(Random Forest - Smooth)")

plt.xlabel('Position level')

plt.ylabel('Salary')

plt.show()

# for 100 trees

from sklearn.ensemble import RandomForestRegressor

regressor = RandomForestRegressor(n\_estimators = 100, random\_state = 0)

regressor.fit(X,y)

#higher resolution graph

X\_grid = np.arange(min(X),max(X),0.01)

X\_grid = X\_grid.reshape(len(X\_grid),1)

plt.scatter(X,y, color='red')

plt.plot(X\_grid, regressor.predict(X\_grid),color='blue')

plt.title("Truth or Bluff(Random Forest - Smooth)")

plt.xlabel('Position level')

plt.ylabel('Salary')

plt.show()

y\_pred=regressor.predict([[6.5]])

y\_pred

# for 300 trees

from sklearn.ensemble import RandomForestRegressor

regressor = RandomForestRegressor(n\_estimators = 300, random\_state = 0)

regressor.fit(X,y)

#higher resolution graph

X\_grid = np.arange(min(X),max(X),0.01)

X\_grid = X\_grid.reshape(len(X\_grid),1)

plt.scatter(X,y, color='red') #plotting real points

plt.plot(X\_grid, regressor.predict(X\_grid),color='blue') #plotting for predict points

plt.title("Truth or Bluff(Random Forest - Smooth)")

plt.xlabel('Position level')

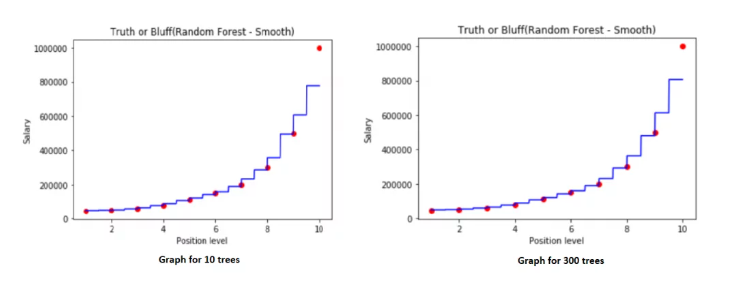
plt.ylabel('Salary')

plt.show()

y\_pred=regressor.predict([[6.5]])

y\_pred

Đầu ra của đoạn mã trên sẽ là đồ thị và giá trị dự đoán. Dưới đây là các biểu đồ:



Biểu đồ đầu ra

**Kết thúc**

Như bạn đã quan sát, mô hình 10 cây dự đoán mức lương cho 6,5 năm kinh nghiệm là 167.000. Mô hình 100 cây dự đoán 158.300 cây và mô hình 300 cây dự đoán 160.333,33. Do đó, số lượng cây càng nhiều, kết quả của chúng tôi càng chính xác.

# TÀI LIỆU THAO KHẢO

<https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/>

[https://www.analyticsvidhya.com/blog/2016/04/tree-based-algorithms-complete- tutorial-scratch-in-python/](https://www.analyticsvidhya.com/blog/2016/04/tree-based-algorithms-complete-%20tutorial-scratch-in-python/)

[R - Random Forest (tutorialspoint.com)](https://www.tutorialspoint.com/r/r_random_forest.htm)

<https://www.r-bloggers.com/2021/04/random-forest-in-r/>

[How to Build Random Forests in R (Step-by-Step) (statology.org)](https://www.statology.org/random-forest-in-r/)

[How to Build Random Forests in R (Step-by-Step) (statology.org)](https://www.statology.org/random-forest-in-r/)

[R Random Forest Tutorial with Example (guru99.com)](https://www.guru99.com/r-random-forest-tutorial.html)