Ta sẽ dùng một ví dụ để giải thích về cây quyết định:

David là quản lý của một câu lạc bộ đánh golf nổi tiếng. Anh ta đang có rắc rối chuyện các thành viên đến hay không đến. Có ngày ai cũng muốn chơi golf nhưng số nhân viên câu lạc bộ lại không đủ phục vụ. Có hôm, không hiểu vì lý do gì mà chẳng ai đến chơi, và câu lạc bộ lại thừa nhân viên.

Mục tiêu của David là tối ưu hóa số nhân viên phục vụ mỗi ngày bằng cách dựa theo thông tin dự báo thời tiết để đoán xem khi nào người ta sẽ đến chơi golf. Để thực hiện điều đó, anh cần hiểu được tại sao khách hàng quyết định chơi và tìm hiểu xem có cách giải thích nào cho việc đó hay không.

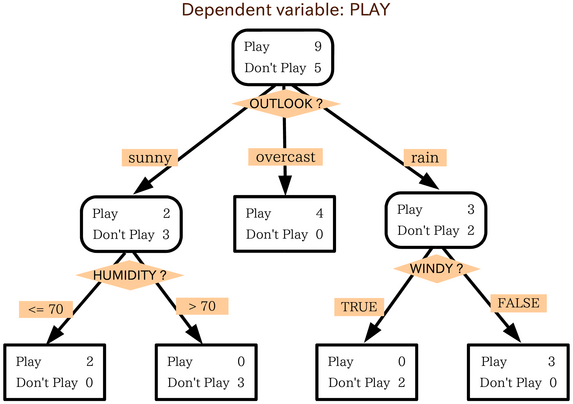
Vậy là trong hai tuần, anh ta thu thập thông tin về:

Trời (*outlook*) (nắng (*sunny*),

Và tất nhiên là số người đến chơi golf vào hôm đó. David thu được một bộ dữ liệu gồm 14 dòng và 5 cột.

[](https://vi.wikipedia.org/wiki/T%E1%BA%ADp_tin:Golf_dataset.png)

Sau đó, để giải quyết bài toán của David, người ta đã đưa ra một mô hình cây quyết định.

[](https://vi.wikipedia.org/wiki/T%E1%BA%ADp_tin:Decision_tree_model.png)

Nhóm người chơi golf khi trời nắng, nhóm chơi khi trời nhiều mây, và nhóm chơi khi trời mưa.

Kết luận thứ nhất: nếu trời nhiều mây, người ta luôn luôn chơi golf. Và có một số người ham mê đến mức chơi golf cả khi trời mưa.

Tiếp theo, ta lại chia nhóm trời nắng thành hai nhóm con. Ta thấy rằng khách hàng không muốn chơi golf nếu độ ẩm lên quá 70%.

Cuối cùng, ta chia nhóm trời mưa thành hai và thấy rằng khách hàng sẽ không chơi golf nếu trời nhiều gió.

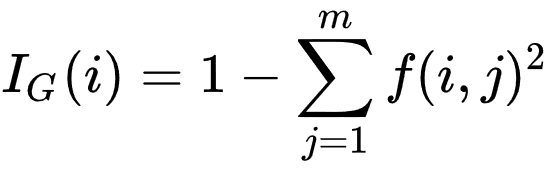
Và đây là lời giải ngắn gọn cho bài toán mô tả bởi cây phân loại. David cho phần lớn nhân viên nghỉ vào những ngày trời nắng và ẩm, hoặc những ngày mưa gió. Vì hầu như sẽ chẳng có ai chơi golf trong những ngày đó. Vào những hôm khác, khi nhiều người sẽ đến chơi golf, anh ta có thể thuê thêm nhân viên thời vụ để phụ giúp công việc.

Kết luận là cây quyết định giúp ta biến một biểu diễn dữ liệu phức tạp thành một cấu trúc đơn giản hơn rất nhiều.

**Gini impurity**

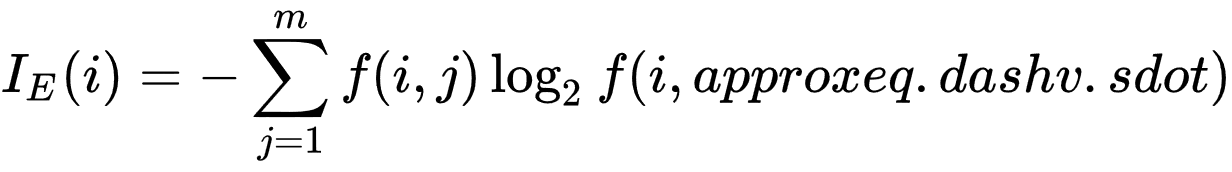
Dùng trong thuật toán CART (Classification and Regression Trees). Nó dựa vào việc bình phương các xác suất thành viên cho mỗi thể loại đích trong nút. Giá trị của nó tiến đến cực tiểu (bằng 0) khi mọi trường hợp trong nút rơi vào một thể loại đích duy nhất.

Giả sử *y* nhận các giá trị trong {1, 2,..., *m*} và gọi *f(i,j)* là tần suất của giá trị *j* trong nút *i*. Nghĩa là *f(i,j)* là tỷ lệ các bản ghi với *y*=*j* được xếp vào nhóm *i*.



### Entropy

Dùng trong các thuật toán sinh cây [ID3](https://vi.wikipedia.org/w/index.php?title=Thu%E1%BA%ADt_to%C3%A1n_ID3&action=edit&redlink=1), [C4.5](https://vi.wikipedia.org/w/index.php?title=Thu%E1%BA%ADt_to%C3%A1n_C4.5&action=edit&redlink=1) và C5.0. Số đo này dựa trên khái niệm [entropy](https://vi.wikipedia.org/wiki/Entropy) trong [lý thuyết thông tin](https://vi.wikipedia.org/wiki/L%C3%BD_thuy%E1%BA%BFt_th%C3%B4ng_tin) (*information theory*).



Cây quyết định là một loại học máy có giám sát (có một biến mục tiêu được xác định trước) hầu hết được sử dụng trong các bài toán phân loại. Nó hoạt động cho cả biến đầu vào và đầu ra phân loại và tiếp tục. Trong kỹ thuật này, tôi sẽ chia tổng thể hoặc mẫu thành hai hoặc nhiều tập hợp đồng nhất (hoặc tập hợp con) dựa trên bộ tách / phân biệt quan trọng nhất trong các biến đầu vào

**Thí dụ:-**

Giả sử chúng ta có một mẫu gồm 30 sinh viên 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 giải trí? Trong bài toán này, chúng ta cần tách biệt những học sinh chơi cricket trong thời gian rảnh rỗi dựa trên biến đầu vào có ý nghĩa cao giữa cả ba.

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



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 đồng nhất tốt nhất. Bây giờ câu hỏi đặt ra là, làm thế nào nó xác định được biến và sự phân chia? Để làm điều này, cây quyết định sử dụng các thuật toán khác nhau

### Các loại cây quyết định

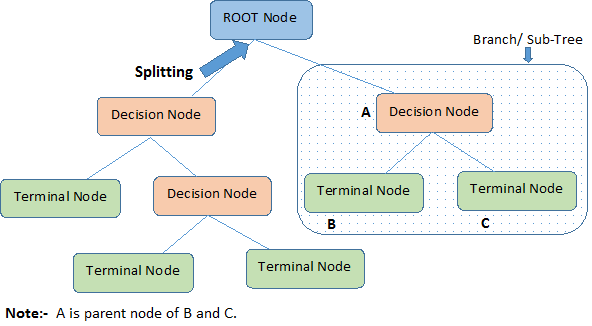
Các loại cây quyết định dựa trên loại biến mục tiêu mà chúng ta có. Nó có thể có hai loại:

1. **Cây quyết định biến phân loại: Cây** quyết định có biến mục tiêu phân loại thì nó được gọi là cây quyết định biến phân loại. Ví dụ: - Trong tình huống trên của bài toán sinh viên, trong đó biến mục tiêu là “Học sinh có chơi cricket hay không” tức là CÓ hoặc KHÔNG.
2. **Cây quyết định biến liên tục: Cây** quyết định có biến mục tiêu liên tục thì nó được gọi là Cây quyết định biến liên tục.

**Ví dụ: -**  Giả sử chúng ta gặp vấn đề trong việc dự đoán liệu một khách hàng có thanh toán phí bảo hiểm gia hạn của mình với một công ty bảo hiểm hay không (có / không). Ở đây chúng ta biết rằng thu nhập của khách hàng là một biến số đáng kể nhưng công ty bảo hiểm không có chi tiết thu nhập cho tất cả khách hàng. Bây giờ, như chúng ta biết đây là một biến số quan trọng, sau đó chúng ta có thể xây dựng một cây quyết định để dự đoán thu nhập của khách hàng dựa trên nghề nghiệp, sản phẩm và nhiều biến số khác. Trong trường hợp này, chúng tôi đang dự đoán các giá trị cho biến liên tục.

### Thuật ngữ quan trọng liên quan đến thuật toán dựa trên cây

Hãy xem các thuật ngữ cơ bản được sử dụng với cây Quyết định:

1. **Nút gốc:** Nó đại diện cho toàn bộ tập hợp hoặc mẫu và nó tiếp tục được chia thành hai hoặc nhiều tập hợp đồng nhất.
2. **Chia tách:** Là một quá trình phân chia một nút thành hai hoặc nhiều nút con.
3. **Nút quyết định:** Khi một nút con tách thành các nút con xa hơn, thì nó được gọi là nút quyết định.
4. **Nút lá / nút đầu cuối:** Các nút không phân chia được gọi là nút Lá hoặc nút đầu cuối.
5. [](https://www.analyticsvidhya.com/wp-content/uploads/2015/01/Decision_Tree_2.png)**Cắt tỉa:** Khi chúng ta loại bỏ các nút con của một nút quyết định, quá trình này được gọi là cắt tỉa. Bạn có thể nói ngược lại quá trình tách.
6. **Cành / Cây** phụ **:** Một phần phụ của toàn bộ cây được gọi là nhánh hoặc cây phụ.
7. **Nút cha và nút con:** Một nút được chia thành các nút con được gọi là nút cha của các nút con trong đó các nút con là con của nút cha.

Đây là những thuật ngữ thường được sử dụng cho cây quyết định. Như chúng ta biết rằng mọi thuật toán đều có ưu điểm và nhược điểm, dưới đây là những yếu tố quan trọng mà bạn nên biết.

### Ưu điểm

1. **Dễ hiểu** : Đầu ra cây quyết định rất dễ hiểu ngay cả đối với những người không có kiến ​​thức về phân tích. Nó không yêu cầu bất kỳ kiến ​​thức thống kê nào để đọc và giải thích chúng. Biểu diễn đồ họa của nó rất trực quan và người dùng có thể dễ dàng liên hệ giả thuyết của họ.
2. **Hữu ích trong việc thăm dò dữ liệu:** Cây quyết định là một trong những cách nhanh nhất để xác định hầu hết các biến quan trọng và mối quan hệ giữa hai hoặc nhiều biến. Với sự trợ giúp của cây quyết định, chúng ta có thể tạo các biến / tính năng mới có khả năng dự đoán biến mục tiêu tốt hơn. Bạn có thể tham khảo bài viết ( [Thủ thuật nâng cao sức mạnh của mô hình hồi quy](https://www.analyticsvidhya.com/blog/2013/10/trick-enhance-power-regression-model-2/) ) để biết một thủ thuật như vậy. Nó cũng có thể được sử dụng trong giai đoạn thăm dò dữ liệu. Ví dụ, chúng tôi đang giải quyết một vấn đề mà chúng tôi có sẵn thông tin về hàng trăm biến, ở đó cây quyết định sẽ giúp xác định biến quan trọng nhất.
3. **Yêu cầu làm sạch dữ liệu ít hơn:** Nó yêu cầu ít làm sạch dữ liệu hơn so với một số kỹ thuật mô hình hóa khác. Nó không bị ảnh hưởng bởi những giá trị ngoại lai và thiếu sót ở một mức độ hợp lý.
4. **Kiểu dữ liệu không phải là một ràng buộc:** Nó có thể xử lý cả biến số và biến phân loại.
5. **Phương pháp phi tham số:** Cây quyết định được coi là một phương pháp phi tham số. Điều này có nghĩa là cây quyết định không có giả định về phân bố không gian và cấu trúc bộ phân loại.

### Nhược điểm

1. **Quá khớp:**  Quá khớp là một trong những khó khăn thực tế nhất đối với các mô hình cây quyết định. Vấn đề này được giải quyết bằng cách thiết lập các ràng buộc về thông số mô hình và cắt tỉa (được thảo luận chi tiết bên dưới).
2. **Không phù hợp với các biến liên tục** : Trong khi làm việc với các biến số liên tục, cây quyết định sẽ mất thông tin khi nó phân loại các biến trong các danh mục khác nhau.

## 2. Cây hồi quy so với cây phân loại

Chúng ta đều biết rằng các nút đầu cuối (hoặc các lá) nằm ở dưới cùng của cây quyết định. Điều này có nghĩa là cây quyết định thường được vẽ ngược sao cho lá là phần dưới cùng và rễ là phần ngọn (hình dưới đây).



Cả hai cây đều hoạt động gần như tương tự nhau, hãy cùng xem xét sự khác biệt và giống nhau cơ bản giữa cây phân loại và cây hồi quy:

1. Cây hồi quy được sử dụng khi biến phụ thuộc là liên tục. Cây phân loại được sử dụng khi biến phụ thuộc có tính phân loại.
2. Trong trường hợp cây hồi quy, giá trị thu được bởi các nút đầu cuối trong dữ liệu huấn luyện là phản hồi trung bình của quan sát rơi vào vùng đó. Do đó, nếu một quan sát dữ liệu không nhìn thấy rơi vào vùng đó, chúng tôi sẽ đưa ra dự đoán của nó với giá trị trung bình.
3. Trong trường hợp cây phân loại, giá trị (lớp) thu được bởi nút đầu cuối trong dữ liệu huấn luyện là phương thức quan sát rơi vào vùng đó. Do đó, nếu một quan sát dữ liệu không nhìn thấy rơi vào vùng đó, chúng tôi sẽ đưa ra dự đoán của nó với giá trị chế độ.
4. Cả hai cây đều phân chia không gian dự đoán (các biến độc lập) thành các vùng riêng biệt và không trùng lặp. Để đơn giản hơn, bạn có thể coi các vùng này là các hộp hoặc hộp có kích thước cao.
5. Cả hai cây đều tuân theo cách tiếp cận tham lam từ trên xuống được gọi là tách nhị phân đệ quy. Chúng tôi gọi nó là 'từ trên xuống' vì nó bắt đầu từ ngọn cây khi tất cả các quan sát có sẵn trong một vùng duy nhất và liên tiếp chia không gian của bộ dự đoán thành hai nhánh mới ở dưới cây. Nó được gọi là 'tham lam' bởi vì, thuật toán quan tâm (tìm kiếm biến tốt nhất có sẵn) chỉ về sự phân tách hiện tại, chứ không phải về sự phân tách trong tương lai sẽ dẫn đến một cây tốt hơn.
6. Quá trình tách này được tiếp tục cho đến khi đạt được tiêu chí dừng do người dùng xác định. Ví dụ: chúng ta có thể yêu cầu thuật toán dừng lại khi số lượng quan sát trên mỗi nút nhỏ hơn 50.
7. Trong cả hai trường hợp, quá trình tách cây dẫn đến các cây phát triển hoàn toàn cho đến khi đạt được các tiêu chí dừng. Tuy nhiên, cây phát triển đầy đủ có khả năng trang bị quá nhiều dữ liệu, dẫn đến độ chính xác kém của dữ liệu không nhìn thấy. Điều này mang lại sự "cắt tỉa". Tỉa là một trong những kỹ thuật được sử dụng để giải quyết việc mặc trang phục quá lố. Chúng ta sẽ tìm hiểu thêm về nó trong phần sau.

## 3. Làm thế nào để các thuật toán dựa trên cây quyết định vị trí phân chia?

Quyết định phân chia chiến lược ảnh hưởng rất nhiều đến độ chính xác của cây. Tiêu chí quyết định khác nhau đối với cây phân loại và cây hồi quy.

Cây quyết định sử dụng nhiều thuật toán để quyết định tách một nút thành hai hoặc nhiều nút con. Việc tạo ra các nút con làm tăng tính đồng nhất của các nút con kết quả. Nói cách khác, chúng ta có thể nói rằng độ tinh khiết của nút tăng lên so với biến đích. Cây quyết định phân chia các nút trên tất cả các biến có sẵn và sau đó chọn sự phân tách dẫn đến hầu hết các nút con đồng nhất.

Việc lựa chọn thuật toán cũng dựa trên loại biến mục tiêu. Hãy xem xét bốn thuật toán được sử dụng phổ biến nhất trong cây quyết định:

### Gini

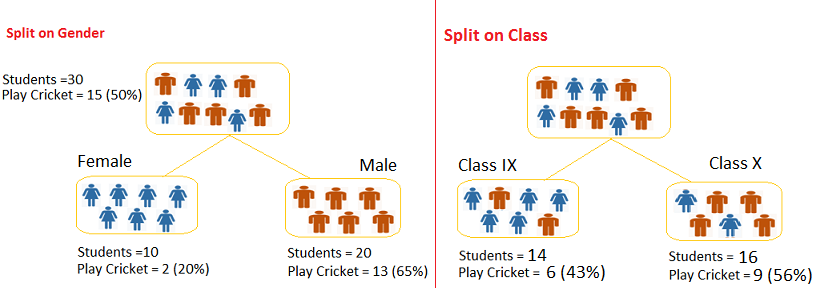
Gini nói, nếu chúng ta chọn ngẫu nhiên hai mục từ một quần thể thì chúng phải cùng loại và xác suất cho điều này là 1 nếu quần thể thuần chủng.

1. Nó hoạt động với biến mục tiêu phân loại "Thành công" hoặc "Thất bại".
2. Nó chỉ thực hiện phân tách nhị phân
3. Giá trị của Gini càng cao thì độ đồng nhất càng cao.
4. CART (Cây phân loại và hồi quy) sử dụng phương pháp Gini để tạo phân tách nhị phân.

**Các bước để tính Gini cho một lần tách**

1. Tính Gini cho các nút con, sử dụng công thức tổng bình phương xác suất thành công và thất bại (p ^ 2 + q ^ 2).
2. Tính Gini cho phần tách bằng cách sử dụng điểm Gini có trọng số của mỗi nút của phần tách đó

**Ví dụ: -** Tham khảo ví dụ đã sử dụng ở trên, chúng tôi muốn tách các học sinh dựa trên biến mục tiêu (chơi cricket hay không). Trong ảnh chụp nhanh bên dưới, chúng tôi phân chia dân số bằng hai biến đầu vào Giới tính và Lớp. Bây giờ, tôi muốn xác định sự phân tách nào đang tạo ra các nút con đồng nhất hơn bằng cách sử dụng Gini.

[](https://www.analyticsvidhya.com/wp-content/uploads/2015/01/Decision_Tree_Algorithm1.png)**Phân chia theo giới tính:**

1. Tính toán, Gini cho nút phụ Nữ = (0,2) \* (0,2) + (0,8) \* (0,8) = 0,68
2. Gini cho nút phụ Nam = (0,65) \* (0,65) + (0,35) \* (0,35) = 0,55
3. Tính Gini có trọng số cho Giới tính phân chia = (10/30) \* 0,68 + (20/30) \* 0,55 = **0,59**

**Tương tự cho Split on Class:**

1. Gini cho nút phụ Lớp IX = (0,43) \* (0,43) + (0,57) \* (0,57) = 0,51
2. Gini cho lớp nút phụ X = (0,56) \* (0,56) + (0,44) \* (0,44) = 0,51
3. Tính toán Gini có trọng số cho Phân loại = (14/30) \* 0,51 + (16/30) \* 0,51 = **0,51**

Ở trên, bạn có thể thấy rằng điểm Gini cho Split on Gender cao hơn Split on Class,  do đó, sự phân chia nút sẽ diễn ra trên Gender.

Bạn có thể thường bắt gặp thuật ngữ 'Tạp chất Gini' được xác định bằng cách lấy giá trị gini trừ đi 1. Vì vậy, về mặt toán học, chúng ta có thể nói,

Tạp chất Gini = 1-Gini

### Chi-Square

Nó là một thuật toán để tìm ra ý nghĩa thống kê giữa sự khác biệt giữa các nút con và nút cha. Chúng tôi đo lường nó bằng tổng bình phương của sự khác biệt tiêu chuẩn hóa giữa tần số quan sát và dự kiến ​​của biến mục tiêu.

1. Nó hoạt động với biến mục tiêu phân loại "Thành công" hoặc "Thất bại".
2. Nó có thể thực hiện hai hoặc nhiều lần tách.
3. Giá trị của Chi-Square càng cao thì ý nghĩa thống kê của sự khác biệt giữa nút con và nút cha càng cao.
4. Chi-Square của mỗi nút được tính bằng công thức,
5. Chi-square = ((Thực tế - Dự kiến) ^ 2 / Dự kiến) ^ 1/2
6. Nó tạo ra cây có tên CHAID (Máy dò tương tác tự động Chi-square)

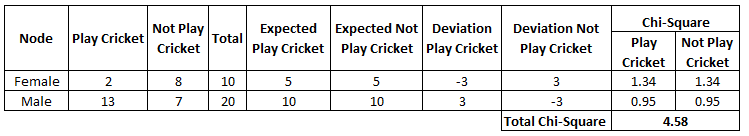
**Các bước để tính Chi-bình phương cho một phép tách:**

1. Tính Chi-square cho từng nút riêng lẻ bằng cách tính toán độ lệch cho cả Thành công và Thất bại
2. Chi-bình phương được tính toán bằng cách sử dụng Tổng của tất cả Chi bình phương của sự thành công và Thất bại của mỗi nút của phép tách

**Ví dụ:** Hãy làm việc với ví dụ trên mà chúng ta đã sử dụng để tính toán Gini.

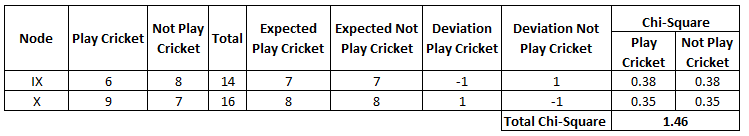
**Phân chia theo giới tính:**

1. Đầu tiên, chúng tôi đang điền cho nút Female, điền giá trị thực tế cho “ **Play Cricket”** và **“Not Play Cricket”** , ở đây lần lượt là 2 và 8.
2. Tính giá trị kỳ vọng cho " **Chơi Cricket"** và " **Không chơi Cricket"** , ở đây nó sẽ là 5 cho cả hai vì nút cha có xác suất là 50% và chúng tôi đã áp dụng cùng một xác suất trên số Nữ (10).
3. Tính toán độ lệch bằng cách sử dụng công thức, Thực tế - Kỳ vọng. Nó dành cho “ **Chơi Cricket”** (2 - 5 = -3) và “ **Không chơi cricket”** (8 - 5 = 3).
4. Tính Chi-bình phương của nút cho “ **Chơi Cricket** ” và “ **Không chơi Cricket** ” bằng công thức có công thức, **= ((Thực tế - Dự kiến) ^ 2 / Dự kiến) ^ 1/2** . Bạn có thể tham khảo bảng dưới đây để tính toán.
5. Làm theo các bước tương tự để tính toán giá trị Chi-square cho nút Nam.
6. Bây giờ, hãy thêm tất cả các giá trị Chi-square để tính Chi-square cho Giới tính đã chia.

[](https://www.analyticsvidhya.com/wp-content/uploads/2015/01/Decision_Tree_Chi_Square1.png)

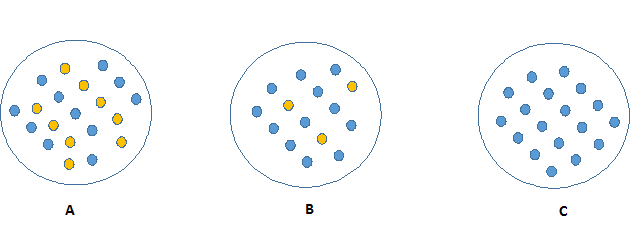
**Phân chia theo lớp:**

Thực hiện các bước tính toán tương tự cho phép chia trên Class và bạn sẽ thấy bảng bên dưới.

[](https://www.analyticsvidhya.com/wp-content/uploads/2015/01/Decision_Tree_Chi_Square_2.png)Ở trên, bạn có thể thấy rằng Chi-square cũng xác định rằng sự phân chia Giới tính có ý nghĩa hơn so với Lớp học.

### Thông tin thu được:

Nhìn vào hình ảnh bên dưới và nghĩ xem nút nào có thể được mô tả dễ dàng. Tôi chắc chắn, câu trả lời của bạn là C vì nó yêu cầu ít thông tin hơn vì tất cả các giá trị đều tương tự nhau. Mặt khác, B yêu cầu nhiều thông tin hơn để mô tả nó và A yêu cầu thông tin tối đa. Nói cách khác, chúng ta có thể nói rằng C là một nút tinh khiết, B ít tạp chất hơn và A không tinh khiết hơn.

[](https://www.analyticsvidhya.com/wp-content/uploads/2015/01/Information_Gain_Decision_Tree2.png)

Bây giờ, chúng ta có thể xây dựng một kết luận rằng nút ít không tinh khiết hơn yêu cầu ít thông tin hơn để mô tả nó. Và, nút không tinh khiết hơn yêu cầu nhiều thông tin hơn. Lý thuyết thông tin là một thước đo để xác định mức độ vô tổ chức này trong một hệ thống được gọi là Entropy. Nếu mẫu là hoàn toàn đồng nhất, thì entropy bằng 0 và nếu mẫu được chia đều (50% - 50%), nó có entropy là một.

Entropy có thể được tính bằng công thức: -Entropy, Cây quyết định

Ở đây p và q là xác suất thành công và thất bại tương ứng trong nút đó. Entropy cũng được sử dụng với biến mục tiêu phân loại. Nó chọn phần tách có entropy thấp nhất so với nút cha và các phần tách khác. Entropy càng nhỏ thì càng tốt.

**Các bước để tính toán entropy cho một phép tách:**

1. Tính toán entropy của nút cha
2. Tính toán entropy của từng nút riêng lẻ của phần tách và tính trung bình có trọng số của tất cả các nút con có sẵn trong phần tách.

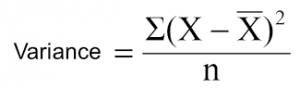
**Ví dụ:** Hãy sử dụng phương pháp này để xác định phép chia tốt nhất cho ví dụ sinh viên.

1. Entropy cho nút cha = - (15/30) log2 (15/30) - (15/30) log2 (15/30) = **1** . Ở đây 1 cho thấy rằng nó là một nút không tinh khiết.
2. Entropy cho nút Nữ = - (2/10) log2 (2/10) - (8/10) log2 (8/10) = 0,72 và cho nút nam, - (13/20) log2 (13/20) - ( 7/20) log2 (7/20) = **0,93**
3. Entropy để phân chia Giới tính = Entropy có trọng số của các nút con = (10/30) \* 0,72 + (20/30) \* 0,93 = **0,86**
4. Entropy cho nút Lớp IX, - (6/14) log2 (6/14) - (8/14) log2 (8/14) = 0,99 và cho nút Lớp X, - (9/16) log2 (9/16) - (7/16) log2 (7/16) = 0,99.
5. Entropy để tách lớp = (14/30) \* 0,99 + (16/30) \* 0,99 = **0,99**

Ở trên, bạn có thể thấy rằng entropy cho  Split on Gender là thấp nhất trong số tất cả, vì vậy cây sẽ phân chia trên  Gender . Chúng ta có thể lấy thông tin thu được từ entropy là **1- Entropy.**

### Giảm phương sai

Cho đến bây giờ, chúng ta đã thảo luận về các thuật toán cho biến mục tiêu phân loại. Phương sai giảm là một thuật toán được sử dụng cho các biến mục tiêu liên tục (bài toán hồi quy). Thuật toán này sử dụng công thức tiêu chuẩn của phương sai để chọn phần tách tốt nhất. Phần tách với phương sai thấp hơn được chọn làm tiêu chí để phân chia tổng thể:

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

Trên thanh X là trung bình của các giá trị, X là thực và n là số giá trị.

**Các bước để tính Phương sai:**

1. Tính phương sai cho mỗi nút.
2. Tính toán phương sai cho mỗi lần tách dưới dạng trung bình có trọng số của phương sai mỗi nút.

**Ví dụ: -** Hãy gán giá trị số 1 để chơi cricket và 0 cho không chơi cricket. Bây giờ hãy làm theo các bước để xác định sự phân chia phù hợp:

1. Phương sai cho nút gốc, giá trị trung bình ở đây là (15 \* 1 + 15 \* 0) / 30 = 0,5 và chúng ta có 15 một và 15 không. Bây giờ phương sai sẽ là ((1-0,5) ^ 2 + (1-0,5) ^ 2 +… .15 lần + (0-0,5) ^ 2 + (0-0,5) ^ 2 +… 15 lần) / 30, điều này có thể được viết là (15 \* (1-0,5) ^ 2 + 15 \* (0-0,5) ^ 2) / 30 = **0,25**
2. Trung bình của nút Nữ = (2 \* 1 + 8 \* 0) /10=0,2 và Phương sai = (2 \* (1-0,2) ^ 2 + 8 \* (0-0,2) ^ 2) / 10 = 0,16
3. Trung bình của Nút nam = (13 \* 1 + 7 \* 0) /20=0,65 và Phương sai = (13 \* (1-0,65) ^ 2 + 7 \* (0-0,65) ^ 2) / 20 = 0,23
4. Phương sai cho giới tính phân chia = Phương sai trọng số của các nút phụ = (10/30) \* 0,16 + (20/30) \* 0,23 = **0,21**
5. Trung bình của nút Lớp IX = (6 \* 1 + 8 \* 0) /14=0,43 và Phương sai = (6 \* (1-0,43) ^ 2 + 8 \* (0-0,43) ^ 2) / 14 = 0,24
6. Trung bình của nút Lớp X = (9 \* 1 + 7 \* 0) /16=0,56 và Phương sai = (9 \* (1-0,56) ^ 2 + 7 \* (0-0,56) ^ 2) / 16 = 0,25
7. Phương sai cho giới tính phân chia = (14/30) \* 0,24 + (16/30) \* 0,25 = **0,25**

Ở trên, bạn có thể thấy rằng phần tách Giới tính có phương sai thấp hơn so với nút mẹ, vì vậy việc tách sẽ diễn ra trên biến Giới tính .

Cho đến đây, chúng tôi đã tìm hiểu về những kiến ​​thức cơ bản về cây quyết định và quá trình ra quyết định liên quan để chọn những phần tách tốt nhất trong việc xây dựng mô hình cây. Như tôi đã nói, cây quyết định có thể được áp dụng cho cả các bài toán hồi quy và phân loại. Chúng ta hãy hiểu những khía cạnh này một cách chi tiết.

## 4. Các tham số chính của thuật toán dựa trên cây là gì và làm thế nào chúng ta có thể tránh quá phù hợp trong cây quyết định?

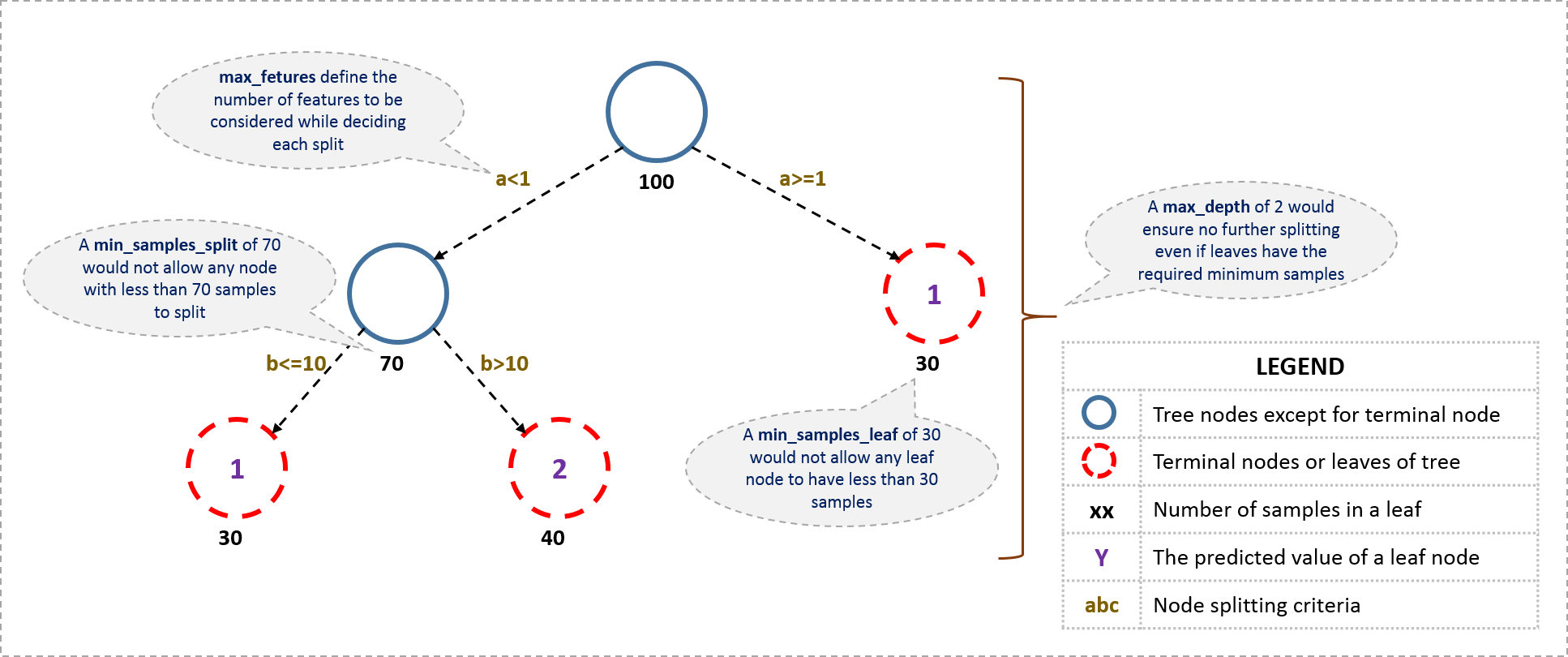
Overfitting là một trong những thách thức chính phải đối mặt khi sử dụng các thuật toán dựa trên cây. Nếu không có tập hợp giới hạn của cây quyết định, nó sẽ cung cấp cho bạn độ chính xác 100% về tập huấn luyện vì trong trường hợp xấu hơn, nó sẽ chỉ tạo ra 1 lá cho mỗi lần quan sát. Do đó, việc ngăn chặn việc trang bị quá mức là quan trọng trong khi lập mô hình cây quyết định và nó có thể được thực hiện theo 2 cách:

1. Đặt các ràng buộc về kích thước cây
2. Tỉa cây

Hãy thảo luận ngắn gọn về cả hai điều này.

### Đặt Ràng buộc trên các thuật toán dựa trên cây

Điều này có thể được thực hiện bằng cách sử dụng các tham số khác nhau được sử dụng để xác định một cây. Đầu tiên, hãy xem cấu trúc chung của cây quyết định:

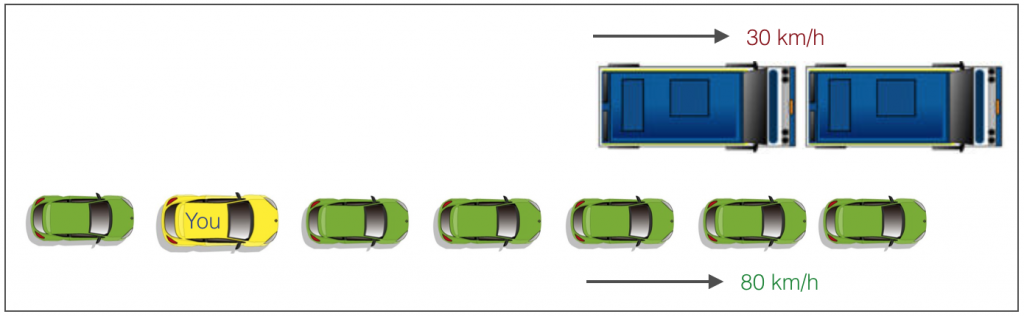
[](https://www.analyticsvidhya.com/wp-content/uploads/2016/02/tree-infographic.png)

Các tham số được sử dụng để xác định cây được giải thích thêm bên dưới. Các thông số được mô tả dưới đây không phụ thuộc vào công cụ. Điều quan trọng là phải hiểu vai trò của các tham số được sử dụng trong mô hình cây. Các tham số này có sẵn trong R & Python.

1. **Các mẫu tối thiểu cho một lần tách nút**
   * Xác định số lượng mẫu (hoặc quan sát) tối thiểu được yêu cầu trong một nút được xem xét để tách.
   * Được sử dụng để kiểm soát việc lắp quá mức. Giá trị cao hơn ngăn cản một mô hình học các quan hệ có thể rất cụ thể đối với mẫu cụ thể được chọn cho một cây.
   * Giá trị quá cao có thể dẫn đến việc không vừa vặn, do đó, cần điều chỉnh bằng cách sử dụng CV.
2. **Các mẫu tối thiểu cho một nút đầu cuối (lá)**
   * Xác định các mẫu (hoặc quan sát) tối thiểu cần thiết trong một nút hoặc lá đầu cuối.
   * Được sử dụng để kiểm soát việc lắp quá mức tương tự như min\_samples\_split.
   * Nói chung, các giá trị thấp hơn nên được chọn cho các vấn đề mất cân bằng về giai cấp vì các vùng mà tầng lớp thiểu số chiếm đa số sẽ rất nhỏ.
3. **Độ sâu tối đa của cây (độ sâu thẳng đứng)**
   * Độ sâu tối đa của cây.
   * Được sử dụng để kiểm soát việc lắp quá mức vì độ sâu cao hơn sẽ cho phép mô hình tìm hiểu các quan hệ rất cụ thể với một mẫu cụ thể.
   * Nên được điều chỉnh bằng cách sử dụng CV.
4. **Số lượng nút đầu cuối tối đa**
   * Số lượng nút hoặc lá đầu cuối tối đa trong một cây.
   * Có thể được định nghĩa thay cho max\_depth. Vì cây nhị phân được tạo nên độ sâu 'n' sẽ tạo ra tối đa 2 ^ n lá.
5. **Các tính năng tối đa cần xem xét để tách**
   * Số lượng các tính năng cần xem xét khi tìm kiếm sự phân chia tốt nhất. Những thứ này sẽ được chọn ngẫu nhiên.
   * Theo quy tắc chung, căn bậc hai của tổng số đối tượng hoạt động tốt nhưng chúng ta nên kiểm tra tối đa 30-40% tổng số đối tượng.
   * Giá trị cao hơn có thể dẫn đến quá khớp nhưng tùy thuộc vào từng trường hợp.

### Cắt tỉa trong thuật toán dựa trên cây

Như đã thảo luận trước đó, kỹ thuật thiết lập ràng buộc là một cách tiếp cận tham lam. Nói cách khác, nó sẽ kiểm tra sự phân tách tốt nhất ngay lập tức và di chuyển về phía trước cho đến khi đạt được một trong các điều kiện dừng được chỉ định. Hãy xem xét trường hợp sau khi bạn đang lái xe:

[](https://www.analyticsvidhya.com/wp-content/uploads/2016/04/graphic.png)

Có 2 làn đường:

1. Một làn đường có ô tô chuyển động với vận tốc 80km / h
2. Một làn đường có xe tải chuyển động với vận tốc 30km / h

Ngay lúc này, bạn là chiếc xe màu vàng và bạn có 2 sự lựa chọn:

1. Rẽ trái và vượt nhanh 2 xe khác
2. Tiếp tục di chuyển trên làn đường hiện tại

Hãy phân tích sự lựa chọn này. Trong lựa chọn cũ, bạn sẽ ngay lập tức vượt xe phía trước và tới phía sau xe tải và bắt đầu di chuyển với tốc độ 30 km / h, tìm kiếm cơ hội để lùi về bên phải. Tất cả các xe ban đầu phía sau bạn sẽ di chuyển phía trước trong khi đó. Đây sẽ là lựa chọn tối ưu nếu mục tiêu của bạn là tối đa hóa khoảng cách được bao phủ trong 10 giây tiếp theo. Trong lựa chọn sau, bạn vượt qua cùng tốc độ, vượt qua xe tải và sau đó vượt có thể tùy thuộc vào tình huống phía trước. Tham lam bạn!



Đây chính xác là sự khác biệt giữa cây quyết định bình thường và cây cắt tỉa. Cây quyết định với các ràng buộc sẽ không nhìn thấy xe tải phía trước và áp dụng cách tiếp cận tham lam bằng cách rẽ trái. Mặt khác, nếu chúng ta sử dụng phương pháp cắt tỉa, thực tế chúng ta sẽ nhìn vào một vài bước trước và đưa ra lựa chọn.

Vì vậy, chúng tôi biết cắt tỉa là tốt hơn. Nhưng làm thế nào để thực hiện nó trong cây quyết định? Ý tưởng là đơn giản.

1. Đầu tiên chúng ta tạo cây quyết định ở độ sâu lớn.
2. Sau đó, chúng tôi bắt đầu ở phía dưới và bắt đầu loại bỏ các lá đang mang lại cho chúng tôi lợi nhuận âm khi so sánh từ phía trên.
3. Giả sử một phép chia đang mang lại cho chúng ta lợi nhuận là -10 (mất 10) và sau đó, lần chia tách tiếp theo cho chúng ta lợi nhuận là 20. Một cây quyết định đơn giản sẽ dừng lại ở bước 1 nhưng khi lược bớt, chúng ta sẽ thấy rằng đạt được là +10 và giữ cả hai lá.

Lưu ý rằng trình phân loại cây quyết định của sklearn hiện không hỗ trợ việc cắt tỉa. Các gói nâng cao như xgboost đã áp dụng việc cắt tỉa cây trong quá trình thực hiện của họ. Nhưng thư viện rpart trong R, cung cấp một chức năng để cắt tỉa. Tốt cho người dùng R!

## 5. Các thuật toán dựa trên cây có tốt hơn mô hình tuyến tính không?

“Nếu tôi có thể sử dụng hồi quy logistic cho các bài toán phân loại và hồi quy tuyến tính cho các bài toán hồi quy, tại sao lại cần sử dụng cây”? Nhiều người trong chúng ta có câu hỏi này. Và, đây cũng là một trong những hợp lệ.

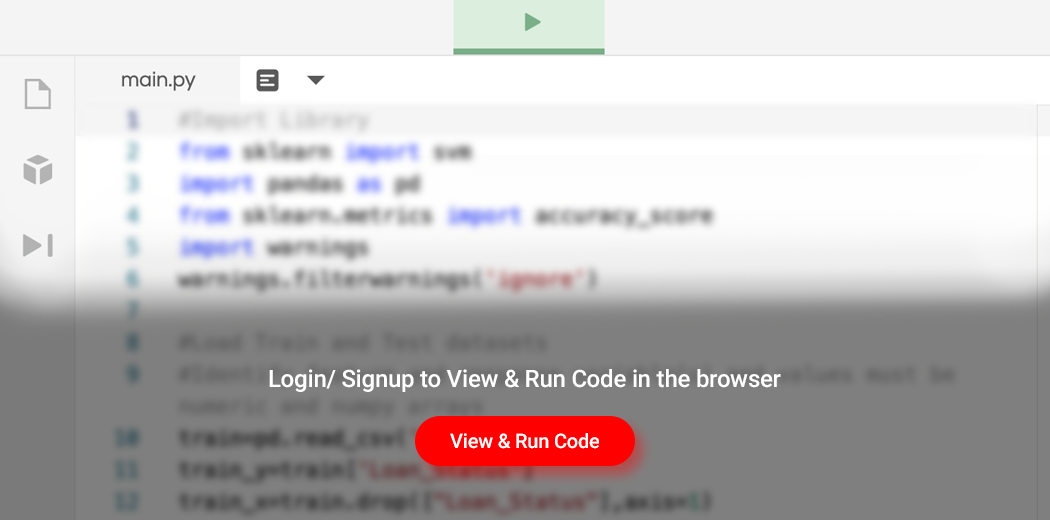
Trên thực tế, bạn có thể sử dụng bất kỳ thuật toán nào. Nó phụ thuộc vào loại vấn đề bạn đang giải quyết. Hãy xem xét một số yếu tố chính sẽ giúp bạn quyết định sử dụng thuật toán nào:

1. Nếu mối quan hệ giữa biến phụ thuộc và biến độc lập được mô hình tuyến tính xấp xỉ tốt thì hồi quy tuyến tính sẽ tốt hơn mô hình dựa trên cây.
2. Nếu có mối quan hệ phức tạp và phi tuyến tính cao giữa các biến phụ thuộc và độc lập, mô hình cây sẽ hoạt động tốt hơn phương pháp hồi quy cổ điển.
3. Nếu bạn cần xây dựng một mô hình dễ giải thích cho mọi người, thì mô hình cây quyết định sẽ luôn hoạt động tốt hơn mô hình tuyến tính. Mô hình cây quyết định thậm chí còn đơn giản hơn để giải thích so với hồi quy tuyến tính!

## 6. Làm việc với các thuật toán dựa trên cây Cây trong R và Python

Đối với người dùng R và người dùng Python, cây quyết định khá dễ thực hiện. Hãy nhanh chóng xem xét bộ mã có thể giúp bạn bắt đầu với thuật toán này. Để dễ sử dụng, tôi đã chia sẻ các mã tiêu chuẩn mà bạn sẽ cần thay thế tên tập dữ liệu và các biến để bắt đầu.

Trên thực tế, bạn có thể xây dựng cây quyết định bằng Python ngay tại đây! Đây là một cửa sổ mã hóa trực tiếp để bạn có thể tìm hiểu mã và tạo ra kết quả:

[](https://id.analyticsvidhya.com/auth/login/?next=https://www.analyticsvidhya.com/blog/2016/04/tree-based-algorithms-complete-tutorial-scratch-in-python/?&utm_source=coding-window-blog&source=coding-window-blog)

Đối với người dùng R, có nhiều gói có sẵn để triển khai cây quyết định như ctree, rpart, tree, v.v.

library(rpart)

> x <- cbind(x\_train,y\_train)

# grow tree

> fit <- rpart(y\_train ~ ., data = x,method="class")

> summary(fit)

#Predict Output

> predicted= predict(fit,x\_test)

Trong đoạn mã trên:

* y\_train - đại diện cho biến phụ thuộc.
* x\_train - đại diện cho biến độc lập
* x - đại diện cho dữ liệu đào tạo.

Đối với người dùng Python, dưới đây là mã:

#Import Library

#Import other necessary libraries like pandas, numpy...

from sklearn import tree

#Assumed you have, X (predictor) and Y (target) for training data set and x\_test(predictor) of test\_dataset

# Create tree object

model = tree.DecisionTreeClassifier(criterion='gini') # for classification, here you can change the algorithm as gini or entropy (information gain) by default it is gini

# model = tree.DecisionTreeRegressor() for regression

# Train the model using the training sets and check score

model.fit(X, y)

model.score(X, y)

#Predict Output

predicted= model.predict(x\_test)

## 7. Các phương pháp tổng hợp trong thuật toán dựa trên cây là gì?

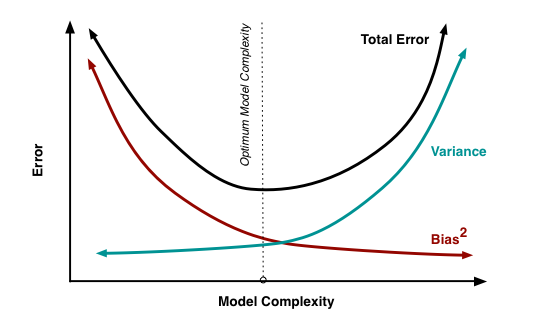
Nghĩa văn học của từ 'quần thể' là nhóm . [Các phương pháp tổng hợp](https://courses.analyticsvidhya.com/courses/ensemble-learning-and-ensemble-learning-techniques?utm_source=blog&utm_medium=tree-based-algorithms-complete-tutorial-scratch-in-python) liên quan đến nhóm các mô hình dự đoán để đạt được độ chính xác và độ ổn định của mô hình tốt hơn. Các phương pháp kết hợp được biết là mang lại hiệu quả tối ưu cho các mô hình dựa trên cây.

Giống như mọi mô hình khác, thuật toán dựa trên cây cũng gặp phải sự sai lệch và phương sai. Bias có nghĩa là 'các giá trị dự đoán khác với giá trị thực tế trung bình bao nhiêu.' Phương sai có nghĩa là 'các dự đoán của mô hình sẽ khác nhau như thế nào tại cùng một điểm nếu các mẫu khác nhau được lấy từ cùng một tổng thể'.

Bạn xây dựng một cây nhỏ và bạn sẽ nhận được một mô hình có phương sai thấp và độ chệch cao. Làm thế nào để bạn quản lý để cân bằng sự cân bằng giữa sự thiên vị và phương sai?

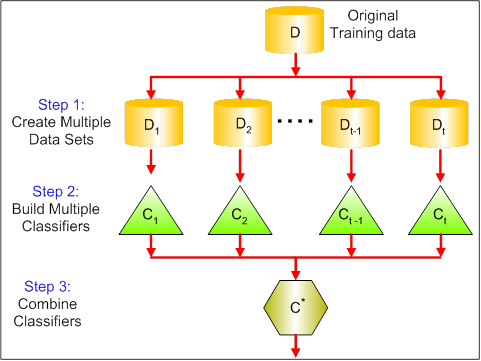
Thông thường, khi bạn tăng độ phức tạp của mô hình, bạn sẽ thấy lỗi dự đoán giảm do độ chệch trong mô hình thấp hơn. Khi bạn tiếp tục làm cho mô hình của mình phức tạp hơn, bạn sẽ kết thúc việc lắp quá mức mô hình của mình và mô hình của bạn sẽ bắt đầu có phương sai cao.

Một mô hình tướng nên duy trì sự cân bằng giữa hai loại lỗi này. Điều này được gọi là **quản lý cân** bằng của các lỗi sai lệch. Học tập theo nhóm là một cách để thực hiện phân tích đánh đổi này.

[](https://www.analyticsvidhya.com/wp-content/uploads/2015/07/model_complexity.png)Một số phương pháp tổng hợp thường được sử dụng bao gồm: Đóng bao, Tăng cường và Xếp chồng. Trong hướng dẫn này, chúng tôi sẽ tập trung chi tiết vào Đóng gói và Tăng tốc.

## 8. Đóng gói là gì? Làm thế nào nó hoạt động?

Phân loại là một [kỹ thuật tổng hợp](https://courses.analyticsvidhya.com/courses/ensemble-learning-and-ensemble-learning-techniques?utm_source=blog&utm_medium=tree-based-algorithms-complete-tutorial-scratch-in-python) được sử dụng để giảm phương sai của các dự đoán của chúng tôi bằng cách kết hợp kết quả của nhiều bộ phân loại được mô hình hóa trên các mẫu con khác nhau của cùng một tập dữ liệu. Hình sau sẽ làm rõ hơn:

[](https://www.analyticsvidhya.com/wp-content/uploads/2015/07/bagging.png)  
Các bước tiếp theo trong đóng bao là:

1. **Tạo nhiều tập dữ liệu** :
   * Việc lấy mẫu được thực hiện với sự thay thế  trên dữ liệu gốc và các tập dữ liệu mới được hình thành.
   * Các tập dữ liệu mới có thể có một phần nhỏ các cột cũng như các hàng, thường là các siêu tham số trong mô hình đóng bao
   * Lấy phân số hàng và cột nhỏ hơn 1 giúp tạo ra các mô hình mạnh mẽ, ít bị trang bị quá mức
2. **Xây dựng nhiều bộ phân loại:**
   * Bộ phân loại được xây dựng trên mỗi tập dữ liệu.
   * Nói chung, cùng một bộ phân loại được mô hình hóa trên mỗi tập dữ liệu và các dự đoán được thực hiện.
3. **Kết hợp các bộ phân loại:**
   * Các dự đoán của tất cả các bộ phân loại được kết hợp bằng cách sử dụng giá trị trung bình, giá trị trung vị hoặc chế độ tùy thuộc vào vấn đề hiện tại.
   * Các giá trị kết hợp thường mạnh mẽ hơn một mô hình duy nhất.

Lưu ý rằng, ở đây số lượng mô hình được xây dựng không phải là một siêu tham số. Số lượng mô hình cao hơn luôn tốt hơn hoặc có thể cho hiệu suất tương tự so với số lượng thấp hơn. Về mặt lý thuyết, có thể chỉ ra rằng phương sai của các dự đoán kết hợp được giảm xuống 1 / n (n: số bộ phân loại) của phương sai ban đầu, theo một số giả định.

Có nhiều cách triển khai các mô hình đóng bao. Rừng ngẫu nhiên là một trong số đó và chúng ta sẽ thảo luận về nó tiếp theo.