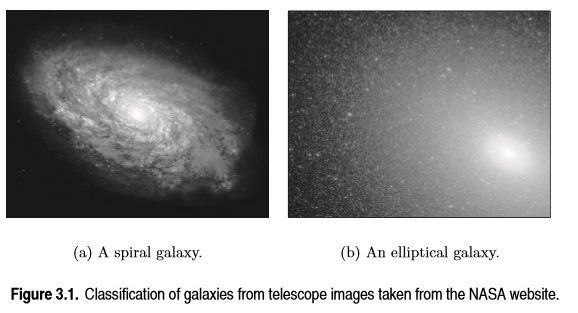
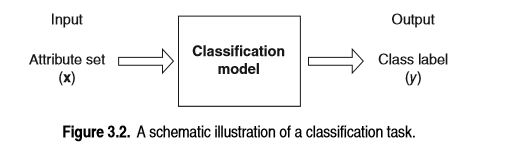
**Classification: các khái niệm và kỹ thuật cơ bản**

Con người có khả năng bẩm sinh để phân loại mọi thứ thành các loại, ví dụ, các nhiệm vụ đơn giản như lọc thư rác hoặc các tác vụ chuyên dụng hơn như nhận dạng các thiên thể trong ảnh kính viễn vọng (xem Hình 3.1). Trong khi phân loại thủ công thường sử dụng các tập dữ liệu nhỏ và đơn giản chỉ có một vài thuộc tính, các tập dữ liệu lớn hơn và phức tạp hơn đòi hỏi một giải pháp tự động.



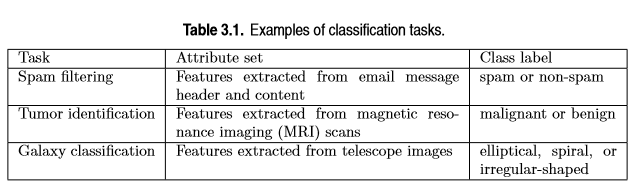


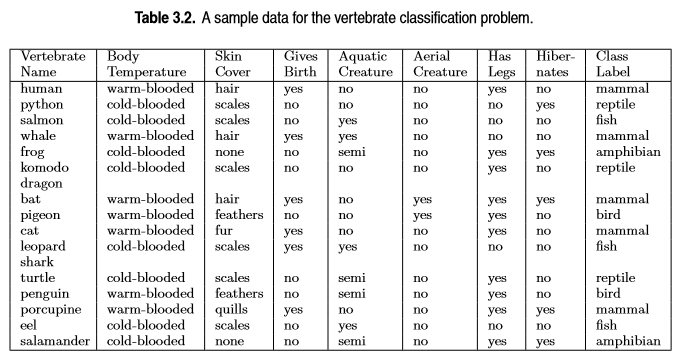
Chương này giới thiệu các khái niệm cơ bản của phân lớp và mô tả một số vấn đề chính của nó, chẳng hạn như model overﬁtting, model selection, and model evaluation. Mặc dù các chủ đề này được minh họa bằng cách sử dụng kỹ thuật phân lớp như là cây quyết định, hầu hết các cuộc thảo luận trong chương này cũng có thể áp dụng cho các kỹ thuật phân lớp khác, nhiều trong số đó được đề cập trong Chương 4.

**3.1 khái niệm cơ bản:**

Hình 3.2 minh họa ý tưởng chung đằng sau phân lớp. Dữ liệu cho một tác vụ phân lớp bao gồm một tập hợp các instances. Mỗi thể hiện như vậy được đặc trưng bởi tuple (x, y), trong đó x là tập hợp các giá trị thuộc tính mô tả thể hiện và y là nhãn lớp của thể hiện. Tập thuộc tính x có thể chứa các thuộc tính thuộc bất kỳ loại nào, trong khi nhãn lớp y phải được phân loại

Một mô hình phân loại là một biểu diễn trừu tượng về mối quan hệ giữa tập thuộc tính và nhãn lớp. Như sẽ thấy trong hai chương tiếp theo, mô hình có thể được biểu diễn theo nhiều cách, ví dụ như dưới dạng cây, bảng xác suất hoặc đơn giản là một vectơ của các tham số có giá trị thực. Chính thức hơn, chúng ta có thể biểu diễn nó một cách toán học như là một hàm mục tiêu f lấy đầu vào là tập thuộc tính x và tạo ra một đầu ra tương ứng với nhãn lớp dự đoán. Mô hình được cho là phân loại chính xác một thể hiện (x, y) nếu f (x) = y.



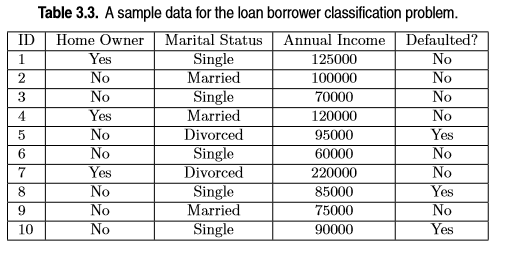


Bảng 3.1 cho thấy các ví dụ về các tập thuộc tính và nhãn lớp cho các tác vụ phân loại khác nhau. Lọc thư rác và nhận dạng khối u là các ví dụ về các vấn đề phân loại nhị phân, trong đó mỗi trường hợp dữ liệu có thể được phân loại thành một trong hai lớp. Nếu số lượng các lớp lớn hơn 2, như trong ví dụ phân loại thiên hà, thì nó được gọi là một vấn đề phân loại đa lớp.

Chúng tôi minh họa các khái niệm cơ bản của phân loại trong chương này với hai ví dụ sau

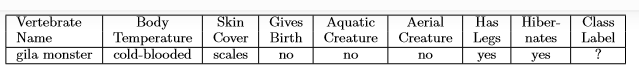
**Ví dụ 3.1. [Phân loại động vật có xương sống]** Bảng 3.2 cho thấy một bộ dữ liệu mẫu để phân loại động vật có xương sống thành động vật có vú, bò sát, chim, cá và lưỡng cư. Bộ thuộc tính bao gồm các đặc điểm của động vật có xương sống như nhiệt độ cơ thể, lớp da và khả năng y. Tập dữ liệu cũng có thể được sử dụng cho một nhiệm vụ phân loại nhị phân, chẳng hạn như phân loại động vật có vú, bằng cách nhóm các loài bò sát, chim, chim và động vật lưỡng cư vào một loại duy nhất được gọi là non-mammals

**Ví dụ 3.2. [Phân loại người vay tiền]** Xem xét vấn đề dự đoán liệu người vay có trả được khoản vay hay không trả được nợ trong các khoản thanh toán cho vay. Tập dữ liệu được sử dụng để xây dựng mô hình phân loại được hiển thị trong Bảng 3.3. Bộ thuộc tính bao gồm thông tin cá nhân của người vay như tình trạng hôn nhân và thu nhập hàng năm, trong khi nhãn lớp cho biết liệu người vay có mặc định về các khoản thanh toán cho vay hay không.



Một mô hình phân loại phục vụ hai vai trò quan trọng trong khai thác dữ liệu. Đầu tiên, nó được sử dụng như một mô hình dự đoán để phân loại các trường hợp chưa được gắn nhãn trước đó. Một mô hình phân loại tốt phải cung cấp dự đoán chính xác với thời gian phản hồi nhanh. Thứ hai, nó phục vụ như một mô hình mô tả để xác định các đặc điểm phân biệt các thể hiện với các lớp khác nhau. Điều này đặc biệt hữu ích cho các ứng dụng quan trọng, chẳng hạn như chẩn đoán y tế, trong đó không cần thiết phải có một mô hình đưa ra dự đoán mà không chứng minh làm thế nào nó đưa ra quyết định như vậy.

Ví dụ, một mô hình phân loại được tạo ra từ tập dữ liệu động vật có xương sống trong Bảng 3.2 có thể được sử dụng để dự đoán nhãn lớp của động vật có xương sống sau:



Ngoài ra, nó có thể được sử dụng như một mô hình mô tả để giúp xác định các đặc điểm của động vật có xương sống như động vật có vú, bò sát, chim, cá hoặc lưỡng cư. Ví dụ, mô hình có thể xác định động vật có vú là động vật có xương sống có máu ấm sinh ra con non.

Có một số điểm đáng chú ý liên quan đến ví dụ trước. Đầu tiên, mặc dù tất cả các thuộc tính được hiển thị trong Bảng 3.2 là định tính, không có hạn chế nào về loại thuộc tính có thể được sử dụng làm biến dự báo. Mặt khác, nhãn lớp phải là loại danh nghĩa. Điều này phân biệt phân loại với các nhiệm vụ mô hình dự đoán khác như hồi quy, trong đó giá trị dự đoán thường là định lượng. Thông tin thêm về hồi quy có thể được tìm thấy trong Phụ lục D.

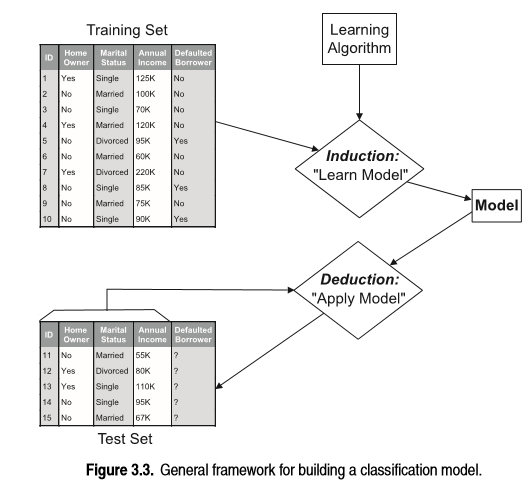
Một điểm đáng chú ý khác là không phải tất cả các thuộc tính có thể liên quan đến nhiệm vụ phân loại. Ví dụ, chiều dài hoặc trọng lượng trung bình của động vật có xương sống có thể không hữu ích để phân loại động vật có vú, vì các thuộc tính này có thể hiển thị cùng một giá trị cho cả động vật có vú và không phải động vật có vú. Một thuộc tính như vậy thường bị loại bỏ trong quá trình tiền xử lý. Các thuộc tính còn lại có thể không thể tự phân biệt các lớp và do đó, phải được sử dụng cùng với các thuộc tính khác. Chẳng hạn, thuộc tính Nhiệt độ cơ thể không cần thiết để phân biệt động vật có vú với các động vật có xương sống khác. Khi nó được sử dụng cùng với Sinh ra, việc phân loại động vật có vú giúp cải thiện đáng kể. Tuy nhiên, khi bao gồm các thuộc tính bổ sung, chẳng hạn như Skin Cover, mô hình sẽ trở nên quá đặc biệt và không còn bao gồm tất cả các động vật có vú. Tìm ra sự kết hợp tối ưu của các thuộc tính phân biệt tốt nhất các thể hiện từ các lớp khác nhau là thách thức chính trong việc xây dựng các mô hình phân loại.

**3.2 General Framework for Classiﬁcation**

Phân lớp là nhiệm vụ gán nhãn cho các trường hợp dữ liệu chưa được gắn nhãn và **classiﬁer** được sử dụng để thực hiện một tác vụ như vậy. Một classiﬁer thường được mô tả theo các mô hình như được minh họa trong phần trước. Mô hình được tạo bằng cách sử dụng một tập hợp các instance đã cho, được gọi là **tập huấn luyện**, chứa các giá trị thuộc tính cũng như nhãn lớp cho mỗi instance. Cách tiếp cận có hệ thống để học một mô hình phân loại được cung cấp một tập huấn luyện được gọi là **thuật toán học tập**. Quá trình sử dụng thuật toán học tập để xây dựng mô hình phân lớp từ dữ liệu đào tạo được gọi là **induction**. Quá trình này cũng thường được mô tả như là "học tập mô hình" hoặc "xây dựng mô hình". Quá trình áp dụng mô hình phân loại này vào các trường hợp thử nghiệm chưa thấy để dự đoán nhãn lớp của chúng được gọi là **khấu trừ**. Do đó, quá trình phân loại bao gồm hai bước: áp dụng thuật toán học tập để đào tạo dữ liệu để học mô hình và sau đó áp dụng mô hình để gán nhãn cho các trường hợp không được gắn nhãn. Hình 3.3 minh họa framework chung cho phân lớp.

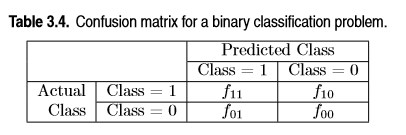
Một kỹ thuật phân lớp đề cập đến một cách tiếp cận chung cho phân lớp, ví dụ, kỹ thuật cây quyết định mà chúng ta sẽ nghiên cứu trong chương này. Kỹ thuật phân loại này giống như hầu hết các kỹ thuật khác, bao gồm một nhóm các mô hình liên quan và một số thuật toán để học các mô hình này. Trong Chương 4, chúng ta sẽ nghiên cứu các kỹ thuật phân loại bổ sung, bao gồm các mạng thần kinh và các máy vectơ hỗ trợ.

Một vài lưu ý về thuật ngữ. Đầu tiên, các thuật ngữ “classifier” và “model” thường được coi là đồng nghĩa. Nếu một kỹ thuật phân loại xây dựng một single, global model thì điều này ổn. Tuy nhiên, trong khi mọi mô hình định nghĩa một classiﬁer, không phải mọi mô hình đều được định nghĩa bởi một classifier duy nhất. Một số classifier , chẳng hạn như k-nearest naibor classifier, không xây dựng một mô hình rõ ràng (Mục 4.3), trong khi các classifier khác, chẳng hạn như các nhóm phân loại, kết hợp đầu ra của một tập hợp các mô hình (Mục 4.10). Thứ hai, thuật ngữ “classifier” thường được sử dụng theo nghĩa chung hơn để chỉ một kỹ thuật phân lớp. Do đó, ví dụ, “classiﬁercây quyết định” có thể đề cập đến kỹ thuật phân lớp cây quyết định hoặc kỹ thuật classiﬁer cây quyết định được xây dựng bằng kỹ thuật đó. May mắn thay, ý nghĩa của các lớp học khác thường rõ ràng từ bối cảnh.

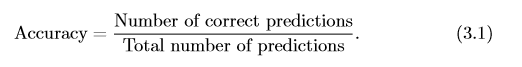


Trong khung chung được chỉ ra trong Hình 3.3, các bước cảm ứng và khấu trừ phải được thực hiện riêng. Trong thực tế, như sẽ được thảo luận sau trong Phần 3.6, các bộ huấn luyện và kiểm tra phải độc lập với nhau để đảm bảo rằng mô hình cảm ứng có thể dự đoán chính xác các nhãn lớp của các trường hợp mà nó chưa từng gặp trước đây. Các mô hình cung cấp những hiểu biết dự đoán như vậy được cho là có hiệu suất khái quát tốt.Hiệu năng của một mô hình (classifier) có thể được đánh giá bằng cách so sánh các nhãn dự đoán với các nhãn thực thể của các thể hiện. Thông tin này có thể được tóm tắt trong một bảng gọi là **ma trận nhầm lẫn**. Bảng 3.4 mô tả ma trận nhầm lẫn cho một vấn đề phân loại nhị phân. Mỗi mục nhập fij biểu thị số lượng phiên bản từ lớp i được dự đoán là của lớp j.

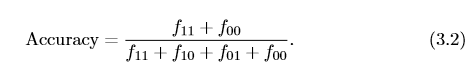
Ví dụ f01 là số lượng trường hợp từ lớp 0 dự đoán không chính xác là lớp 1. Số lượng dự đoán chính xác tạo bởi model là (f11 + f00) và số lượng dự đoán không chính xác là (f10 + f01)



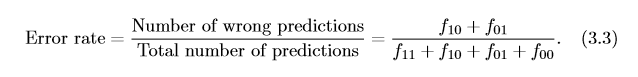
Mặc dù ma trận nhầm lẫn cung cấp thông tin cần thiết để xác định mô hình phân loại hoạt động tốt như thế nào, việc tóm tắt thông tin này thành một số giúp thuận tiện hơn khi so sánh hiệu suất tương đối của các mô hình khác nhau. Điều này có thể được thực hiện bằng cách sử dụng một số liệu đánh giá, chẳng hạn như độ chính xác, được tính theo cách sau:



Đối với các vấn đề phân loại nhị phân, độ chính xác của mô hình được đưa ra bởi:



**Tỷ lệ lỗi** là một số liệu liên quan khác, được xác định như sau đối với các vấn đề phân loại nhị phân:

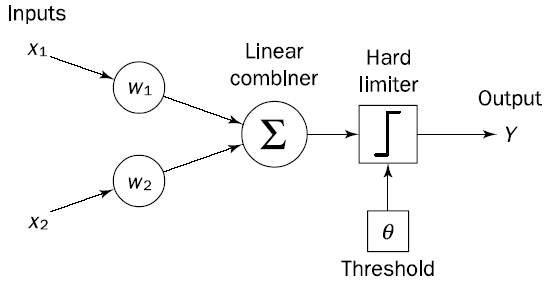


Các thuật toán học tập của hầu hết các kỹ thuật phân loại được thiết kế để học các mô hình đạt độ chính xác cao nhất hoặc tương đương, tỷ lệ lỗi thấp nhất khi áp dụng cho tập kiểm tra. Chúng tôi sẽ xem xét lại chủ đề đánh giá mô hình trong Phần 3.6.

**Perceptron**

1. **Cấu trúc**

Perceptron Là mạng neuron nhân tạo đơn giản nhất gồm một neuron với các trọng số có thể điều chỉnh được và một giới hạn cứng.



1. **Mã giả**

* *Khởi tạo:* (w1, w2, …, wn, t) [-0.5, 0.5]

Trong đó : w1, w2,…, wn là các trọng số khởi tạo ngẫu nhiên với miền giá trị -0.5 đến 0.5.

* *Kích hoạt:* tính kết xuất thực sự tại lần lặp p = 1

Y(p) là kết xuất thực tại bước p, là ngưỡng hoạt hoá.

* *Cập nhật trọng số:*

Trong đó:

* là tốc độ học
* e(p) là sai số ở bước p, tính bởi e(p) = Yd(p)- Y(p) với Yd(p) là kết xuất mong muốn, Y(p) là kết xuất thực tại bước p.
* *Lặp:* tăng p lên 1 và lặp lại bước 2.

**Mã giả:**

/\* initialization \*/

w[n] = [w1,w2, …, wn] ;

x[n] = [[x1,x2, …, xn]1, [x1,x2, …, xn]2, …, [x1,x2, …, xn]p];

theta = abc;

alpha = xyz;

yd = [yd1, yd2, …, ydp]; // collection of yd

y = 0;

continueLearning = true;

for ( i = 0 ; i < n ; i++ ) w[i] = random(-0.5,0.5) ; // random value from -0.5 to 0.5