Classification là 1 model dùng để phân loại data thành vào các class khác nhau qua các ràng buộc.

Để làm bài toán Classification thì cần có training data để máy học trước.

**Classification – Quá trình :**

1. Chuẩn bị training data và rút trích các đặc trưng (**feature** extraction).

Đây là input để tìm ra mô hình của bài toán. Nên ta cần chọn ra những đặc trưng tốt (good feature) của dữ liệu, lược bỏ các feature ko tốt gây nhiễu.

1. Xây dựng mô hình classifier từ tập training data. Bước này thường được gọi là training.

Mục đích là để tìm ra hàm f(x) để gán nhãn (**label**) cho input

f(x) = y với x là các feature hay input đầu vào, y là label cho output.

Để xây dựng mô hình phân lớp này, ta thường sử dụng các thuật toán học giám sát (supervised learning) như KNN, Neural Network, SVM, Decision Tree, Navie Bayers.

1. Kiểm tra dữ liệu với mô hình (make prediction).

Đưa các dữ liệu mới vào model ở trên.

1. Đánh giá model và chọn ra model tốt nhất

Đánh giá mức độ lỗi của dữ liệu testing và training thông qua mô hình tìm được. Nếu ko đạt, ta thay đổi các tham số (turning parameter) của các thuật toán để tìm ra mô hình tốt hơn

Trong các giải thuật phân loại chuẩn (standard classification learning algorithms), các dữ liệu sau khi phân loại thường bị lệch về phía các Majority class (Negative class) và vì thế sẽ xuất hiện nhiều hơn sự phân loại nhầm cho các dữ liệu lẽ ra phải thuộc về Minority class (possitive class).

**Các giải thuật :**

* Perceptron**[1]** (Phát minh bởi *Frank Rosenblatt* vào năm 1957, trước khi Internet ra đời)
* K-nearest Neighbors (Famous Algorithm)**[1]**

Cách đơn giản nhất để phân loại 1 đối tượng chưa được gán nhãn là so sánh các thuộc tính của nó với các đối tượng đã được gián nhãn xem có bằng nhau ko. Nhưng trên thực tế rất hiếm trường hợp 2 đối tượng giống hệt nhau, nên K-nearest (K hàng xóm gần nhất). Chúng ta dùng chỉ số **Distance** để so sánh độ giống nhau giữa 2 đối tượng, thường ngta dùng khoảng cách Euclidean.

Sau khi tính ra **Distance** giữa đối tượng chưa được gán nhãn đó với tất cả các đối tượng trong tập **Training Data**, ta sẽ lấy ra **K** kết quả. Trong trường hợp tốt nhất, **K** kết quả đều dẫn đến 1 class thì tất nhiên đối tượng này sẽ thuộc class đó. Trong trường hợp xấu nhất, **K** kết quả thuộc **K** lớp khác nhau, ta sẽ xem class nào có số lượng lớp nhất và gán class đó cho đối tượng này.

* Random Forest (**RF**) (Breiman 2001)**[1]**

Dùng 1 chuỗi các thuật toán **Decision Tree**, các thuật toán **Decision Tree** này có thể được hiện thực theo 2 cách. Cách 1 : Hỏi các chuyên gia trong lĩnh vực đó. Cách 2 : Dùng tập **Training Data** để tự xây dựng **Decision Tree**.

**RF** luôn bắt đầu với 1 cây decision rỗng, chỉ có 1 điểm bắt đầu và dẫn tới khung trả lời. Tiếp theo, **RF** sẽ tìm ra câu hỏi đầu tiên tốt nhất để bắt đầu. Để tìm ra câu hỏi tốt nhất, đầu tiên nó sẽ hỏi tất cả cấc câu có thể hỏi( có thể lên tới hàng triệu), ứng với mỗi câu hỏi, **RF** sẽ tính toán **Information Gain** để đánh giá xem câu hỏi có tốt ko.

Các cây decision được tạo ra theo cùng 1 cách, do đó sẽ có trường hợp các cây giống hệt nhau. Để tránh trường hợp này, **RF** sẽ xóa ngẫu nhiên 1 vài đối tượng và nhân bản 1 vài đối tượng khác, những thay đổi nhỏ này sẽ dẫn tới việc kết quả thay đổi, tránh trường hợp các cây decision giống nhau, và quá trình này được gọi là **Bootstrapping.**

Ngoài ra để đảm bảo các cây decision có sự khác biệt, **RF** sẽ loại bỏ có chủ đích 1 vài câu hỏi khi xây dựng cây decision. Trong trường hợp này, nếu câu hỏi tốt nhất không được hỏi trong cây decion đó, các câu hỏi khác sẽ được chọn để tạo ra cây. Quá trình này được gọi là **Attribute Sampling.**

* Roughly balanced bagging (Hido & Kashima 2008)
* C4.5
* SVM
* Apriori
* AdaBoost

**Giải thích key words :**

Supervised classification : phân loại có hướng dẫn (có training data)

**[1]** [**https://tech.3si.vn/2016/03/31/ml-classification-part-3/**](https://tech.3si.vn/2016/03/31/ml-classification-part-3/)

[**http://tech.3si.vn/2016/03/28/ml-classification-part-2/**](http://tech.3si.vn/2016/03/28/ml-classification-part-2/)

[**http://tech.3si.vn/2016/03/28/ml-classification-part-1/**](http://tech.3si.vn/2016/03/28/ml-classification-part-1/)