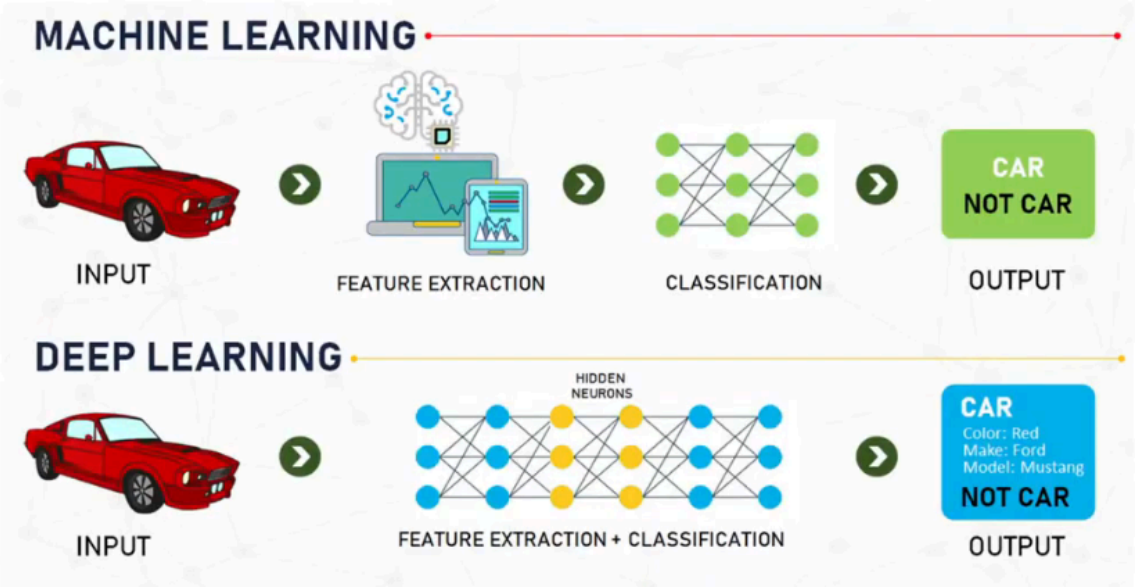
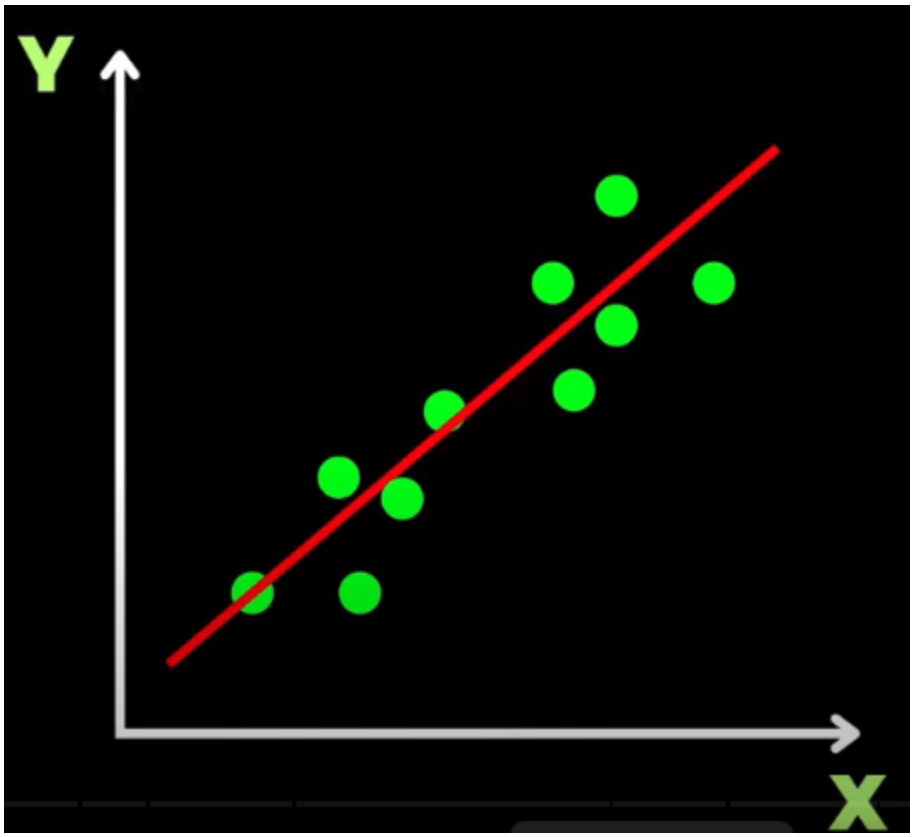
ex

Sự kahcs nhau ML và DL



Thuật toán ML	Mô tả ngắn gọn	Dễ hiểu là...
Linear Regression	Dự đoán giá trị liên tục (vd: giá nhà)	Kéo một đường thẳng tốt nhất qua dữ liệu
Logistic Regression	Dự đoán phân loại nhị phân (vd: spam/không spam)	Trả về xác suất
Decision Tree	Cây quyết định	Hỏi-đáp kiểu "nếu... thì..."
Random Forest	Tập hợp nhiều cây quyết định	Bỏ phiếu để ra kết quả tốt hơn
KNN (K-Nearest Neighbors)	Phân loại dựa trên hàng xóm gần nhất	"Xem hàng xóm là ai rồi bắt chước"
SVM (Support Vector Machine)	Phân loại bằng cách vẽ đường ranh giới tối ưu	Ngăn cách dữ liệu rõ nhất có thể
Naive Bayes	Phân loại dựa trên xác suất Bayes	Giả định các đặc trưng là độc lập

Linear Regression: Mô hình có gắng tìm ra mối quan hệ tuyến tính của đầu vào x và đầu ra ý. Nghĩa là có gắng tìm 1 phương trình y = ax + b để biểu diễn dự đoán



Nếu chỉ 1 đầu vào X thì gọi là simple linear regresssion

Nếu có nhiều hơn 1 đầu vào x thì gọi là multiple linear regression: Y = W1X1 + W2X2....

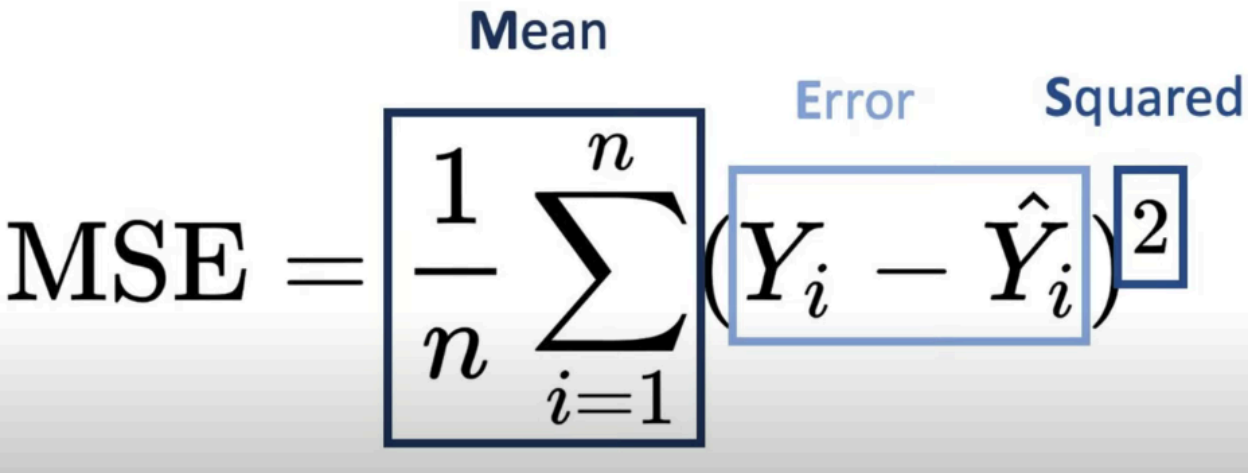
$$y = \sum_{i=0}^{n} w_i x_i$$

x : có thể gọi feature / Independence variable

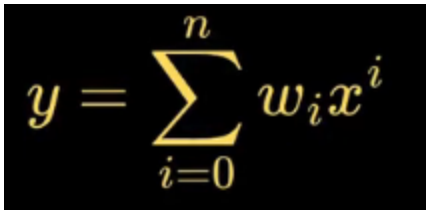
y: có thể gọi là depencedent variable / target

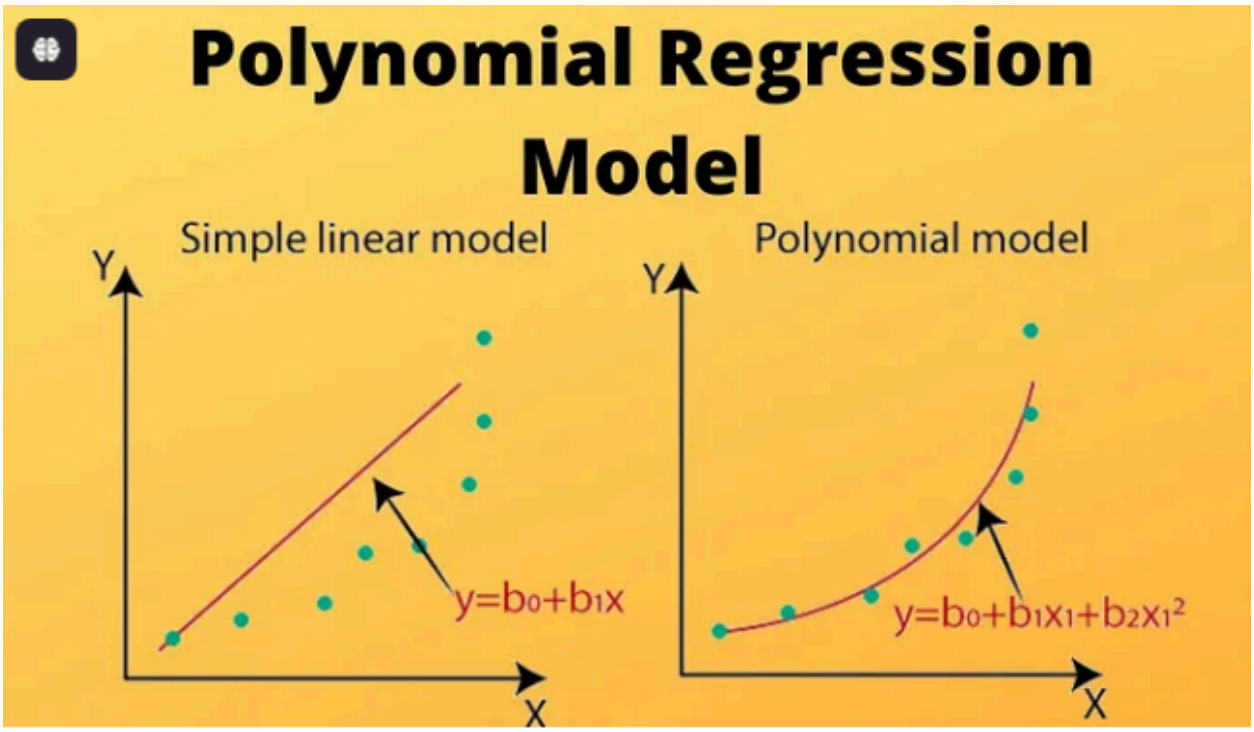
⇒ tìm được y = ax + b sao cho khoảng cách trung bình đến các điểm dữ liệu là nhỏ nhất.

Lost function hay hàm mất mát sẽ công thức MSE :

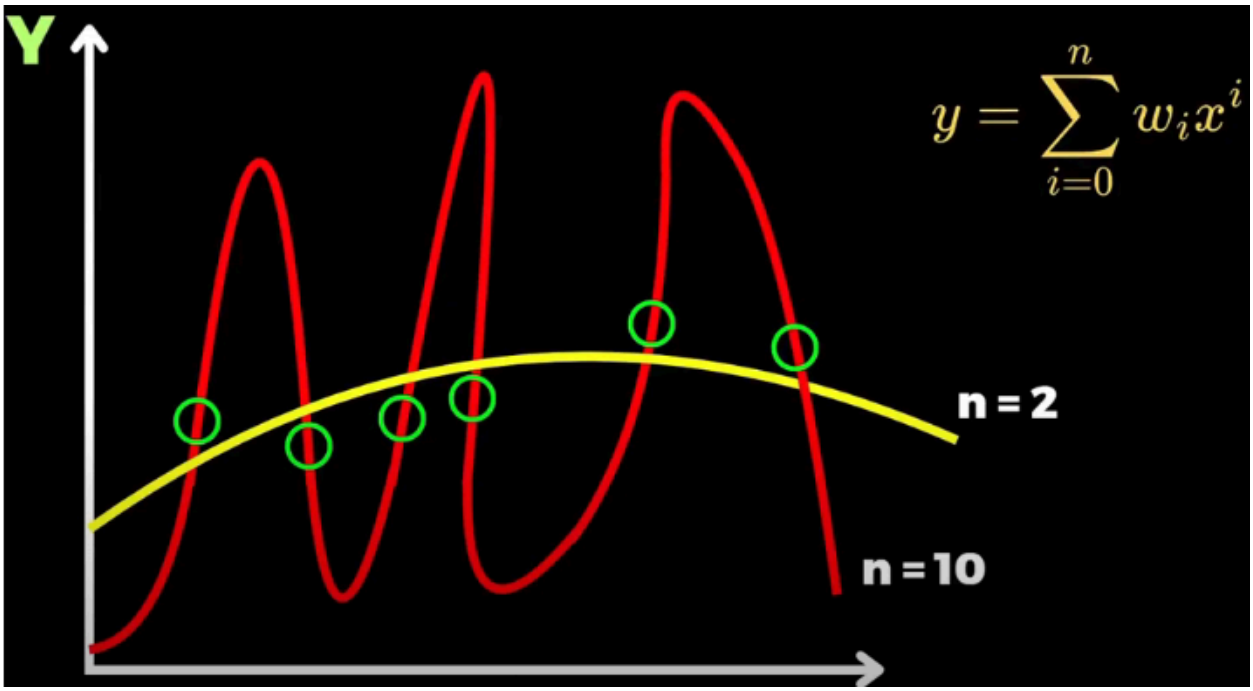


Trong trường hợp bộ dữ liệu phức tạp thig LN sẽ có nhược điểm là có thể vẽ dc nhiều đường thẳng khác nhau. Trong trường hợp này thì sẽ dùng Plolynomial Regresion - hồi quy đa thức

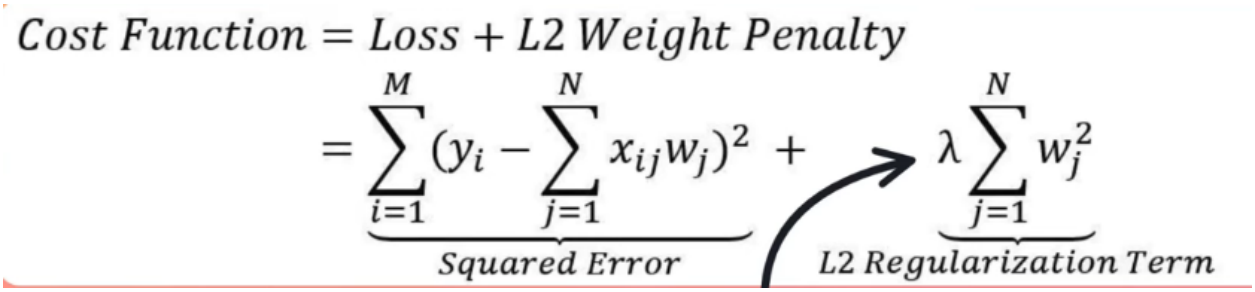




Vấn đề độ phức tạp của mô hình lớn hơn nhiều so với độ phức tạp của dữ liệu ⇒ vdễ xảy ra vấn đề overfitting , nghãi là với dữ liệu huấn luyện thì độ chính xác cao nhưng nếu khi chạy thật thì kém



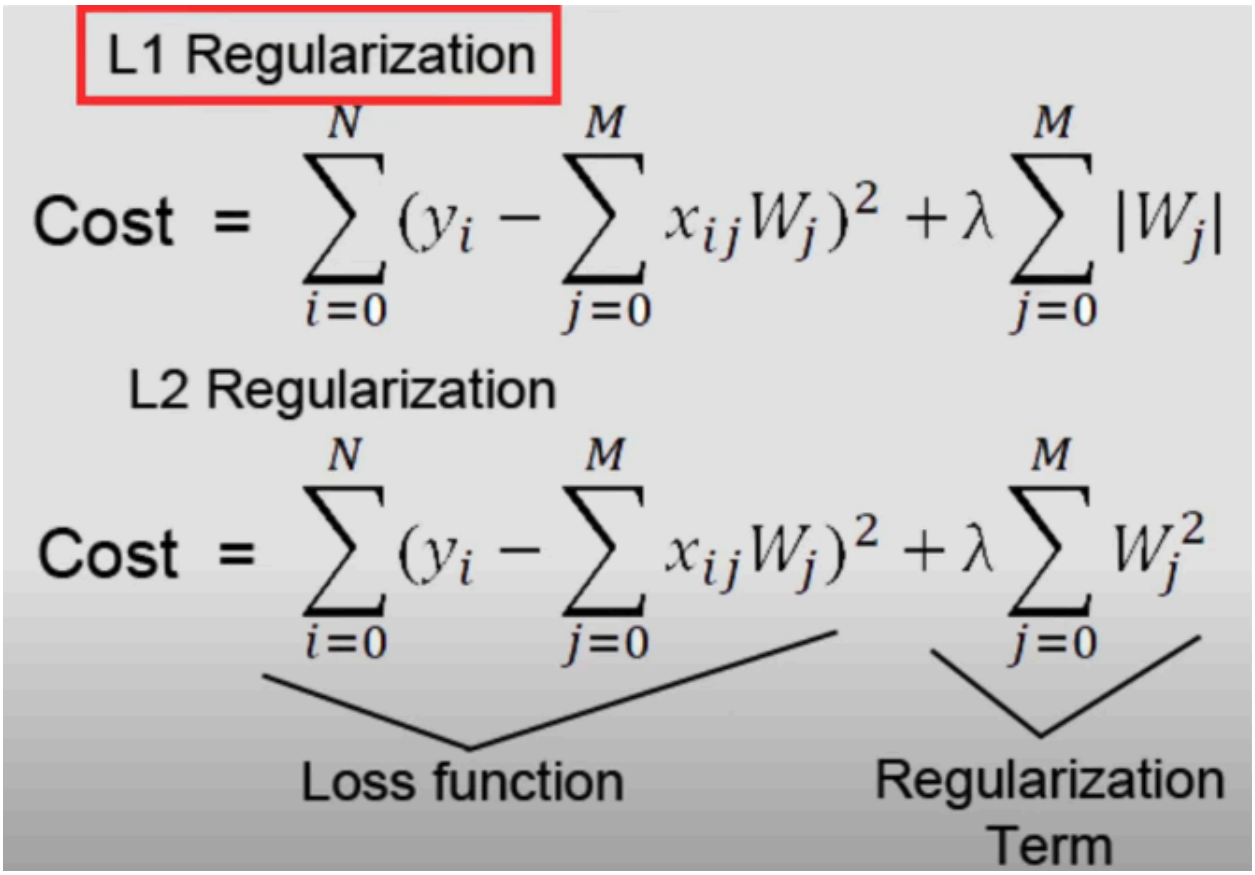
Để giả quyết vấn đề này thì tìm cách vừa tối ưu sai số giữa dự đoán và thực tế cũng như kiểm soát để tối thiểu hóa độ phức tạp của thuật toán ⇒ Regularization - Chính quy khóa

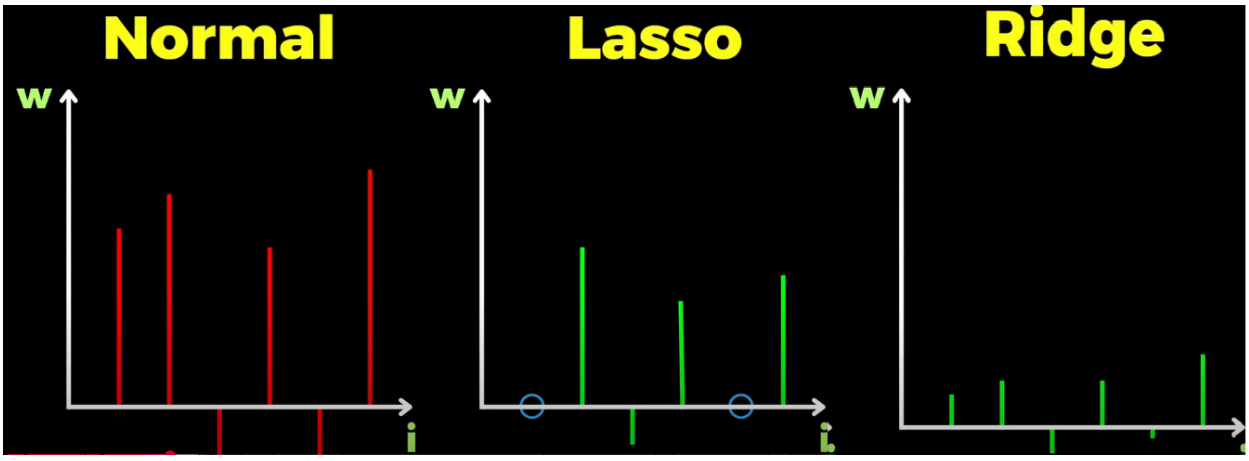


Thông thường thì đối tượng tối ưu là loss function

2 thành phần sẽ đối lập nhau, nếu wj lớn thì lost thất nhưng regularization cao

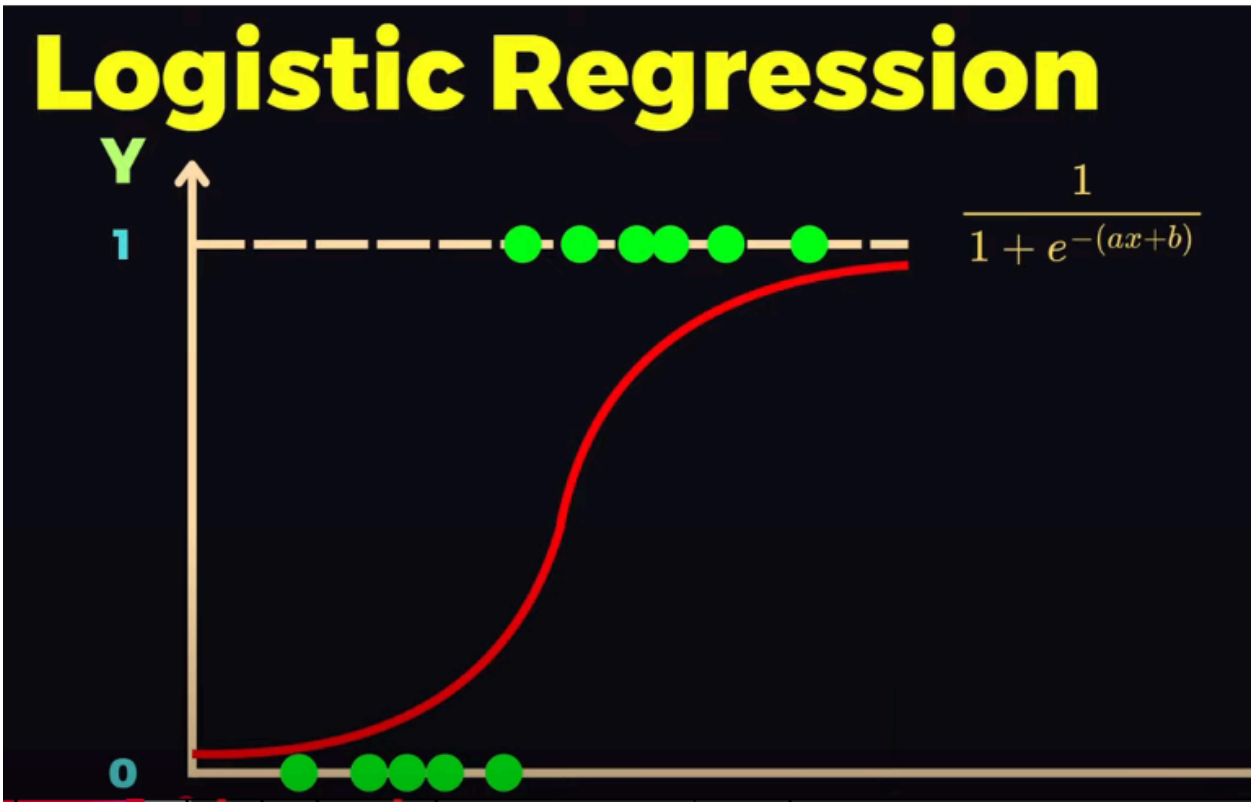
- Có 3 kỹ thuật regularization chính là: L1 là Lasso / L2 gọi là Ridge / Elastic Net (kết hợp L1 và L2)
- Lasso thường được dùng trong Feature Selection(chọn lọc đặng trưng) - vì nó có xu hướng ép cho giá trị cảu 1 vào trọng số trở về 0, khi đó các đặc trưng đó sẽ không còn ảnh hưởng tới mô hình nữa
- Ridge - có xu hướng giảm độ lớn của các trọng số ⇒ giảm ảnh hưởng của các đặc trưng tới dự đoán của mô hình chứ ko triệt tiêu



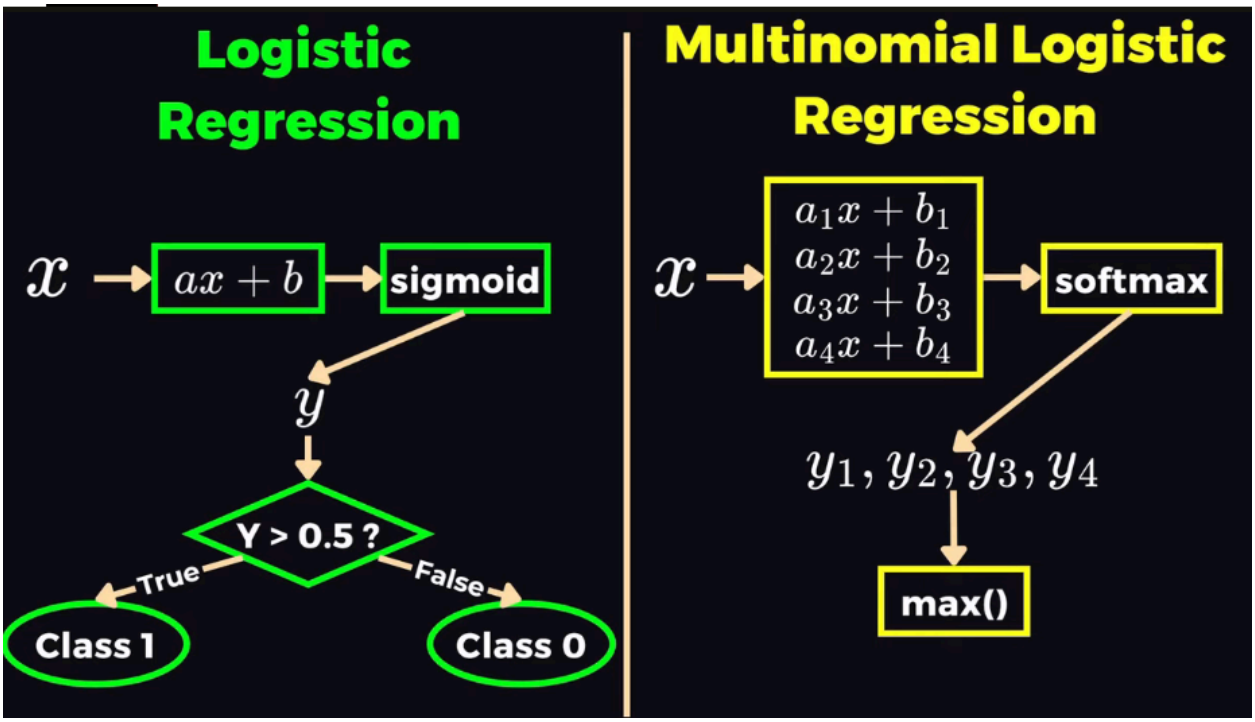


Với LR thì dự đoán của mô hình là -vô cùng đến + vô cùng mà ko có giới hạn

⇒ nếu muốn áp dụng vào bài toán phân loại thì làm sao? ⇒ ánh xạ giá trị từ - đến + về khoảng 0 đến 1 và đặt ra 1 ngưỡng



Nếu bài toán phân loại có nhiều hơn 2 class thì làm thế nào ⇒ Multinomial Logistic Regression

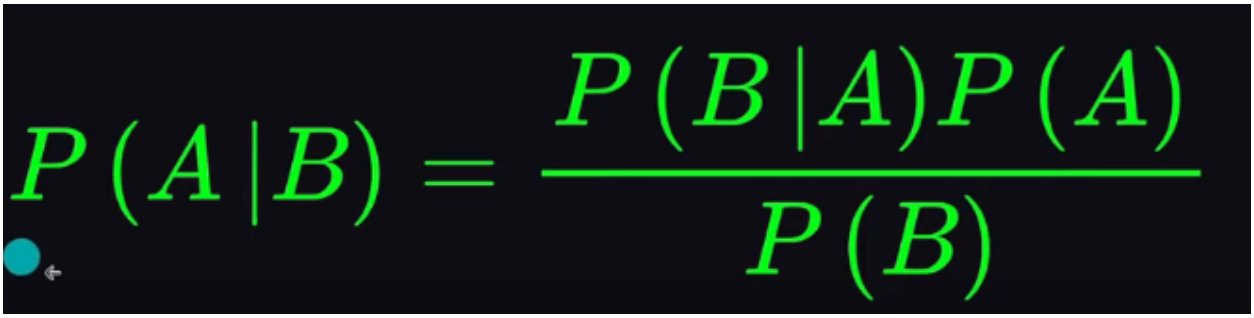


Hàm softmax có tác dụng ánh xạ vector là các phần tử có giá trị nằm trong - + thành các phần tử có giá trị từ 0 đến 1 sao cho tổng các phần tử từ 0 đến 1 (có thể xem là xác xuất)

Định lý bayes

Tính được xác xuất xảy ra 1 sự kiện nào đó biết rằng sự kiện khác đã xảy ra rồi

Xacx xuất xảy ra sự kiện A biết rằng B đã xảy ra bằng xác xuất xảy ra sự kiện B biết ràng A đã xảy ra * xác xuất xảy ra sự kiện A / xác xuất xảy ra sự kiện B

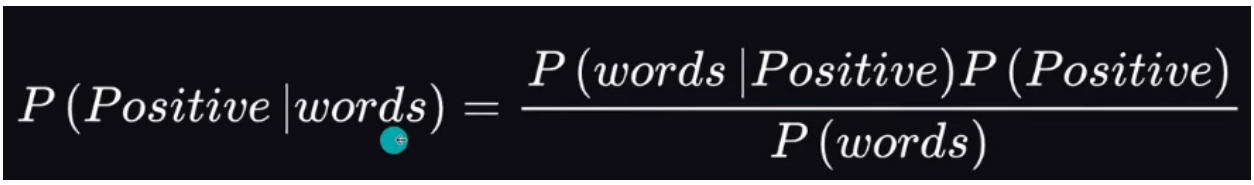


$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Ví dụ muốn xây dự một đoạnvăn bản có ý nghĩa tích cự hay tiêu cực

Bạn code tệ quá.

Xác xuất đoạn text có ý nghĩa tích cực biết rằng nó có chuwacs các từ trên = xác xuất đoạn text có chứa những từ trên có ý nghĩa tích cực * xác xuất đoạn text có ý nghĩa tích cực / xác xuất đoạn text có chứa những từ đó



$$P(Positive|words) = \frac{P(words|Positive)P(Positive)}{P(words)}$$

P(words|positive) = P(bạn, code,tệ, quá| Posotive) = P(bạn|positive)* P(code| bạn, positive)*P(tệ| bạn, code, Positive)* P(quá | bạn, code, tệ, positive)

⇒ vì việc tính toán xác xuất cảu các thần phần sau khó nên người ta sẽ thêm 1 giả thiết " trong 1 đoạn text bất kỳ, nếu chúng ta biết đoạn text này thuộc về là class nào đó (tích cực/tiêu cực) thì xác xuất để các từ khác nhau xuất hiện là hoàn toàn đọc lập với nhau.

⇒

Thuật toán Naive Bayes:

P(words|positive) = P(bạn, code,tệ, quá| Posotive) = P(bạn|positive)* P(code| positive)*P(tệ| Positive)* P(quá | positive)

Multinomial naive bayes: nếu input đầu vào là các biến rời rạc và giá trị là số lần xuất hiện trong đoạn test

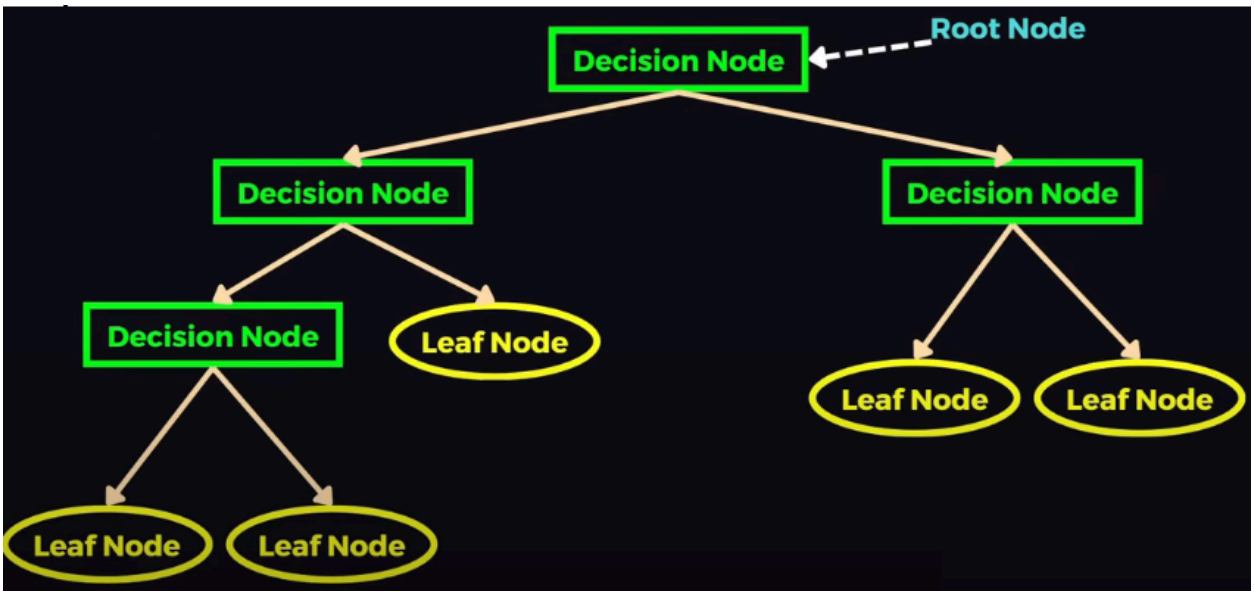
Bernoulli navice bayes: nếu input đầu vào là các biến rời rạc và giá trị là đoạn text đó có xuất hiện hay ko

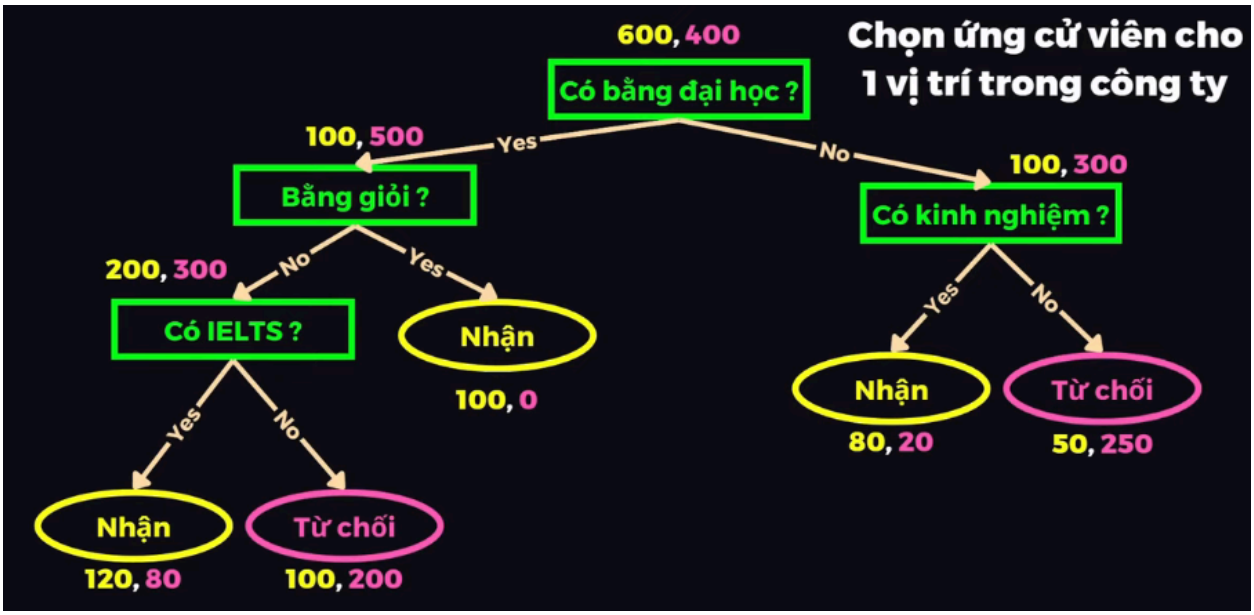
Word	Present
the	1
boy	1
girl	0
sitting	1
standing	0
to	1

Gaussian Naive Bayes: nếu input đầu vào là 1 biến liên tục(âm thanh...) và thêm giả thuyết về phân phối chuẩn

—ÁP DỤNG CHO CẢ HỒI QUY VÀ PHÂN LAOIJF

Decision Tree

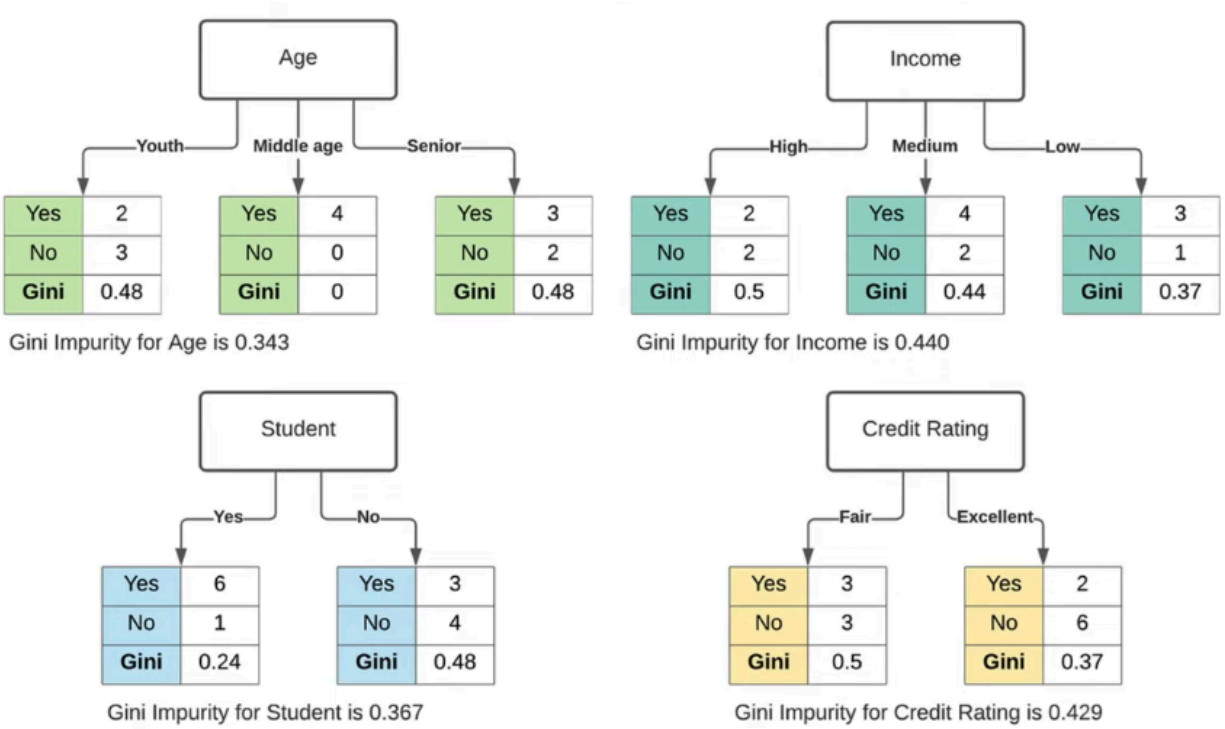




Độ sau của cây, nốt là có bao nhiêu phân tử, độ lệch cảu node lá?

Chọn featuare nào cho Desision Node? Một trong những cách phổ biến là dùng Gini Impurity (C là ssos lượng class). Nếu featurea mà phân bố đều thì Gini cao

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$



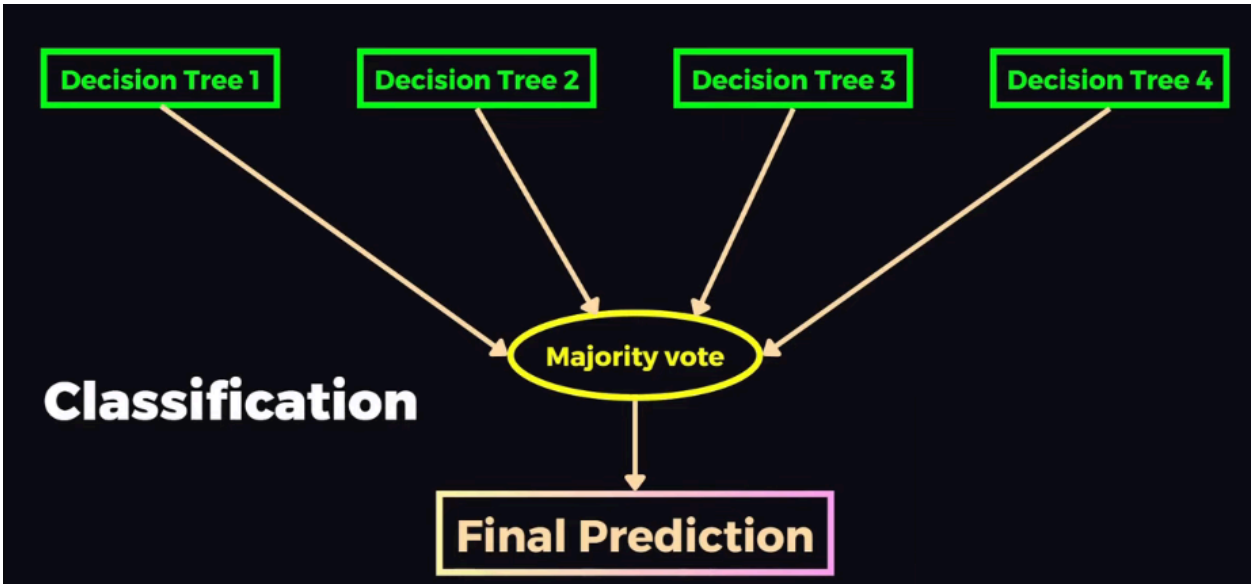
$$Gini = 1 - (0.4^2 - 0.6^2) = 0.48$$

Ngoài ra ta có thể sửa dụng Entropy

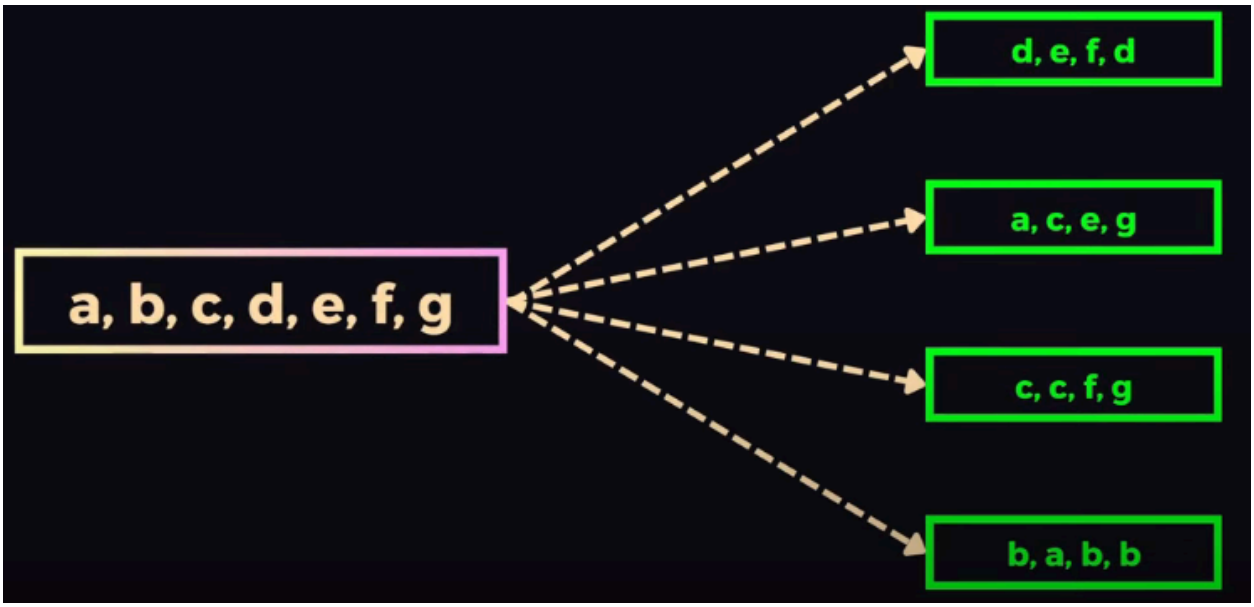
$$Entropy = \sum_{i=1}^{c} -P_i * log_2(P_{i)}$$

⇒ Nhược điểm rất dễ overfitting khi nhiều level và chỉ 1 điểm thay đổi nhỏ cũng có thể thay đổi kết quả.

Trong thực tế sẽ tìm cách kết hợp nhiều Desision Tree lại với nhau để tăng độ chính xác của kết quả ⇒ radom forest ()

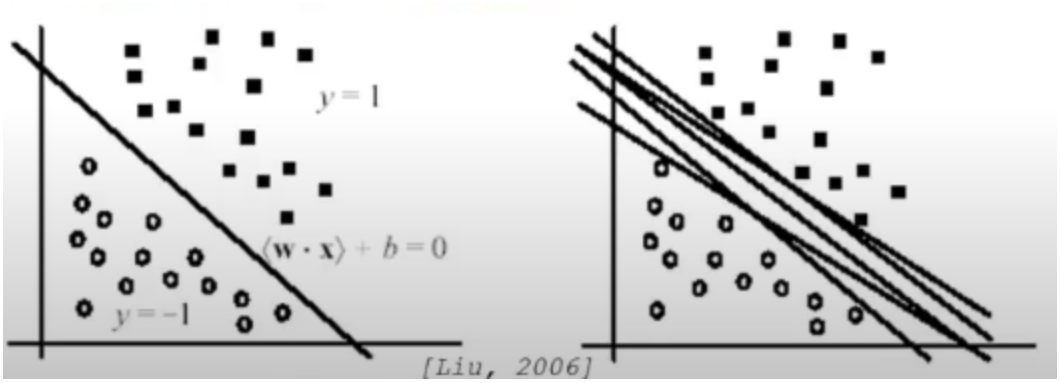


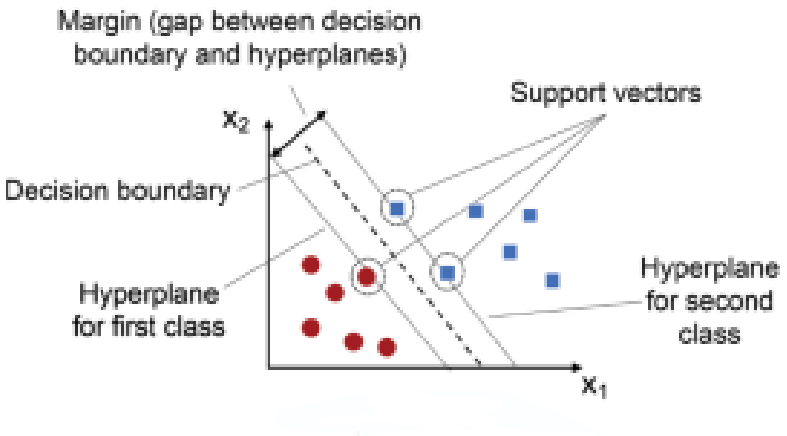
mỗi DT sẽ đươợc huấn luyện trên bộ dữ liệu khác nhau ⇒ sử dụng kỷ thuật **Boostrapping** để tạo ra các bộ dữ liệu khác nhau từ 1 tranning data set

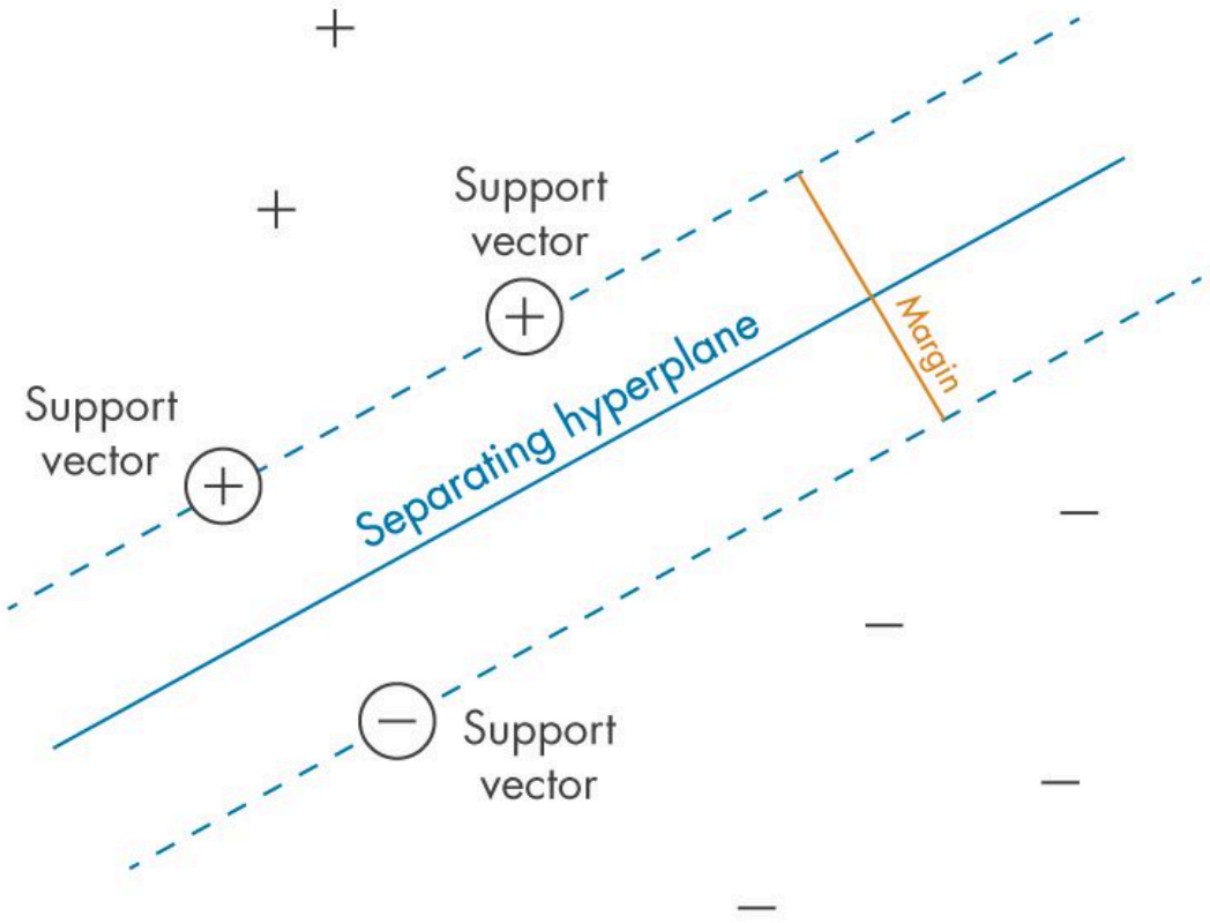


Thuật toán SVM

Mục tiêu của SVM cho bài toán phân loại là tìm ra đường phân cách tốt nhất để có thể phân chia dữ liệu thành các class khác nhau trong không gian đa chiều. Nó sẽ cố gắng tìm ra 1 siêu phẳng (hyperplan) để phân chia dữ liệu. V**ậy chọn line nào? Chọn hàm có hyperplan có max margin**





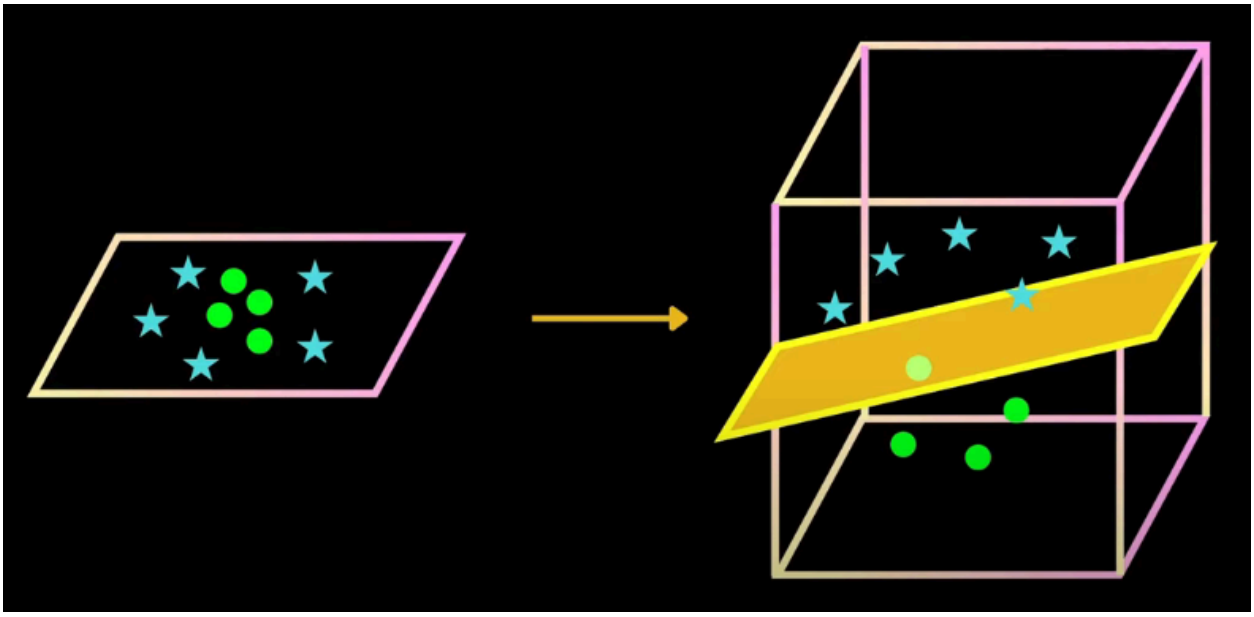


- Tức là **không có đường nào chia tách được hoàn hảo**. ⇒ thiết lập 1 siueeu tham số C để quy định ngưỡng Cho phép **một số điểm vi phạm biên (margin)**,

tức là **cho sai một chút**, nhưng vẫn cố gắng **tối đa hóa margin** và **giảm sai sót tổng thể**.

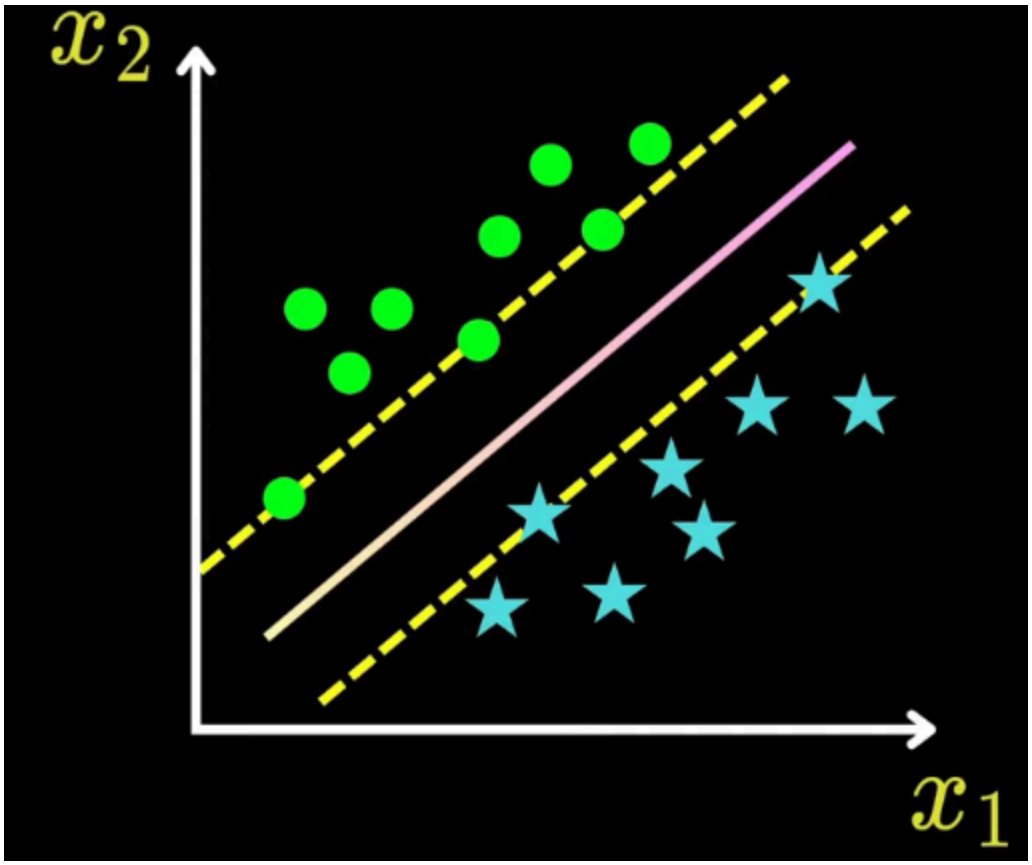
⇒ Khi đó, **việc tìm một đường cách đều là không khả thi nữa** – vì sẽ luôn có điểm nằm sai phía.

Nó làm việc với hàm tuyến tình, đối với các hàm phi tuyến thì dùng kỹ thuật kernel để chuyển đổi ⇒ dùng **Kernel Trick**: **Biến đổi dữ liệu sang không gian khác** (VD từ 2D sang 3D) để **phân tách được tuyến tính ở đó** → sau đó áp dụng lại SVM như bình thường.



Đường ở giữa là đường phân cách tối ưu , vì nó thảo mãn điều kiện cách đều các điểm gần nhất thuộc về 2 class (khoảng cách này gọi là margin) và margin do đường này tạo ra là tối đa.

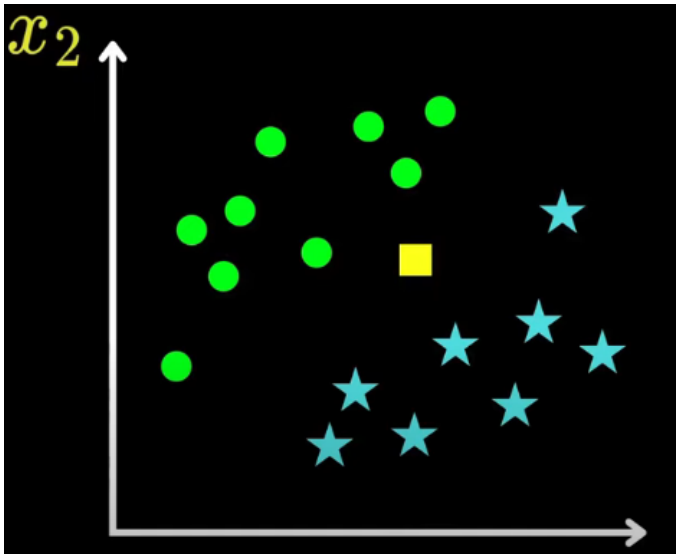
Các đươngg gần với đường phân cách gọi là các support vector: là thành phần quyết định vị trí của đường phân cách tối ưu

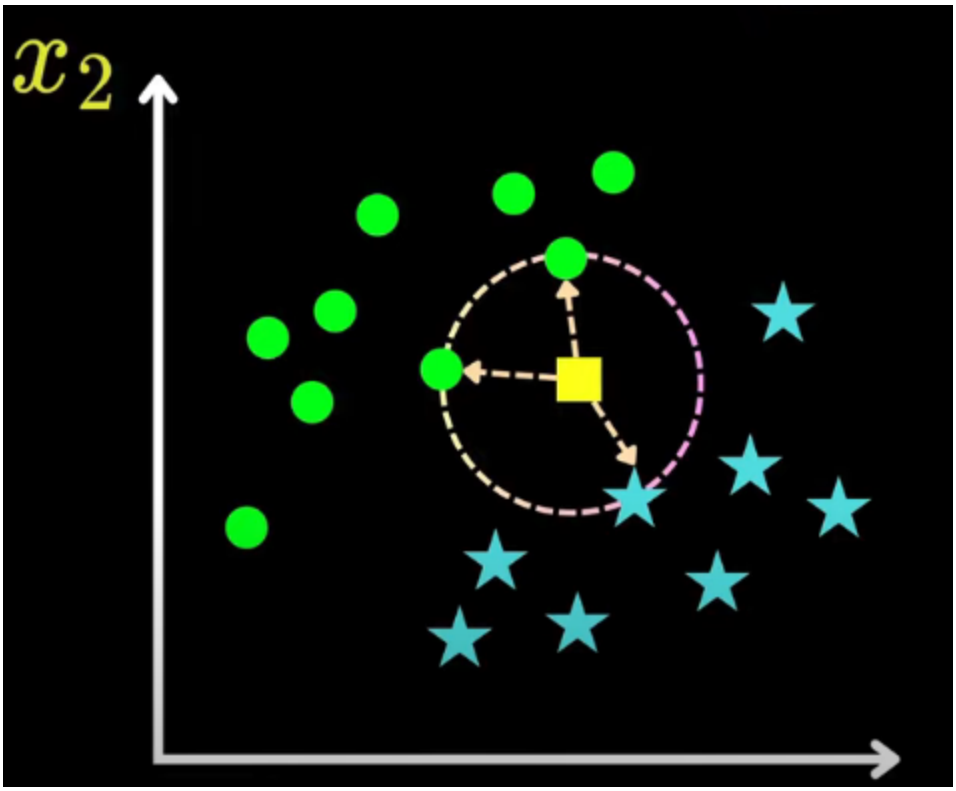


SVM chậm khi quá nhiều datapoint

K-Nearest Neighbors : Không có quán trình huấn luyện

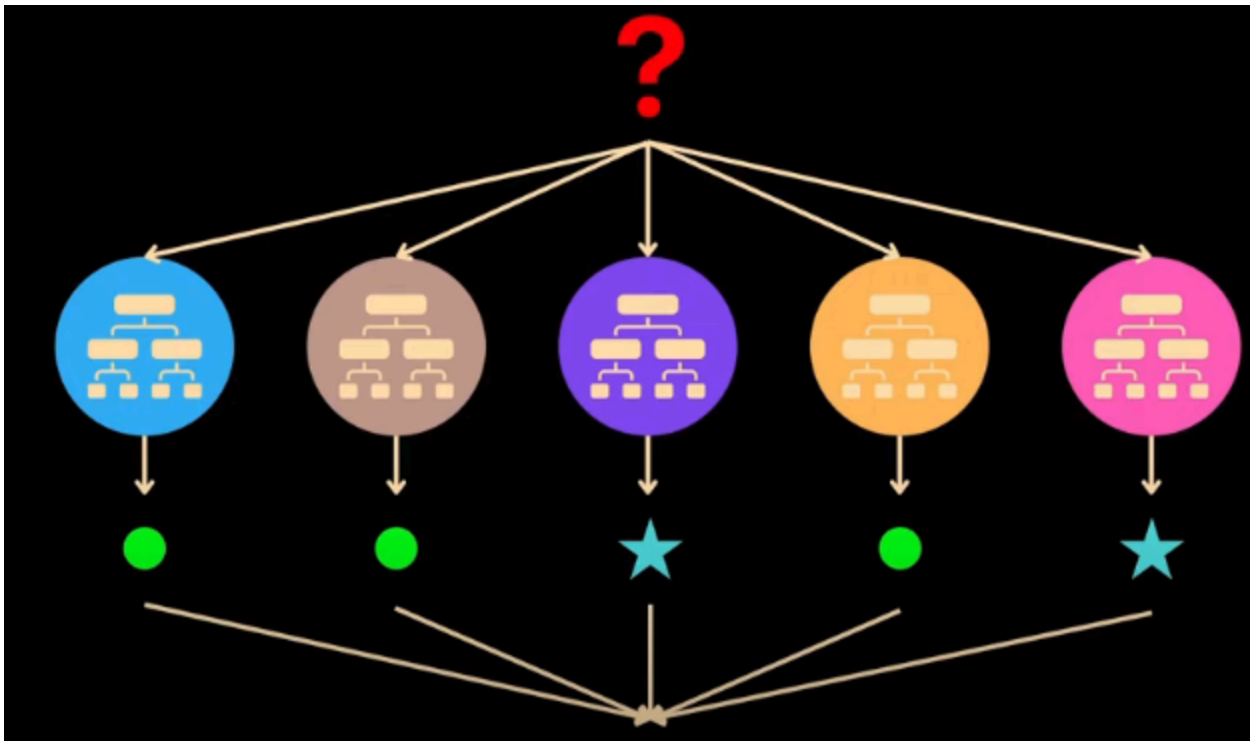
- Cần xác định K ⇒ việc chọn K là quan trọng. Vô cùng chậm khi dữ liệu lớn





Cách kết hợp mô hình đơn giản chó độ chính xác thấp thành mô hình phức tạp có độ chính xác cao gọi là Ensemble Leaning (học tổng hợp)

- Bagging: Nhiều mô hình con khác nhau thường là cùng loại, bộ dữ liệu khác nhau dựa vào boottraping.



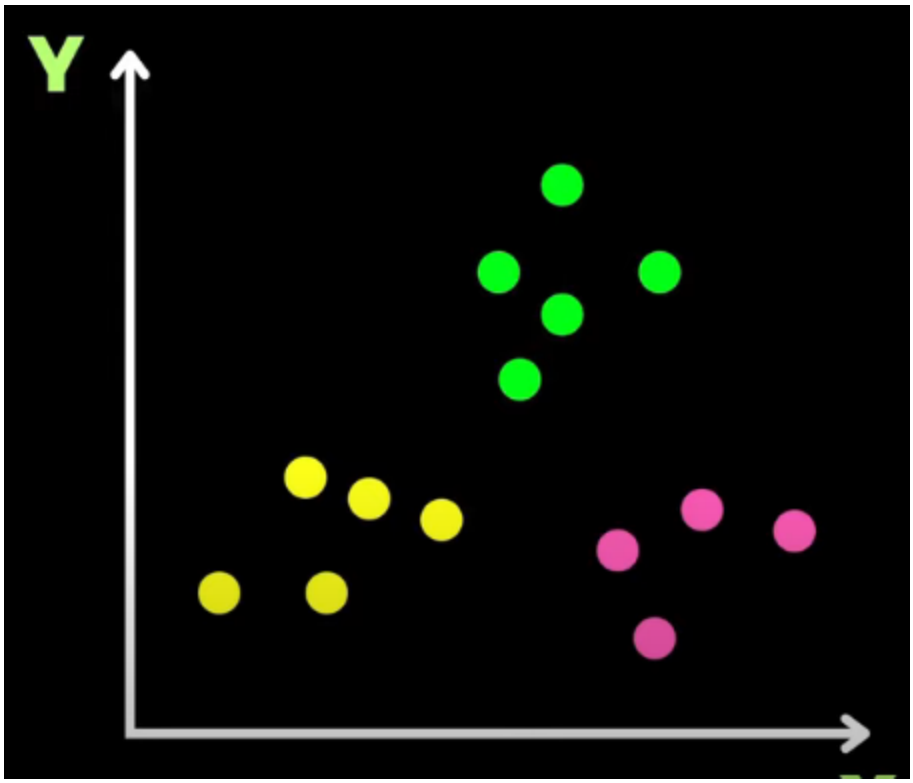
- Boosting
- Voting
- Stacking

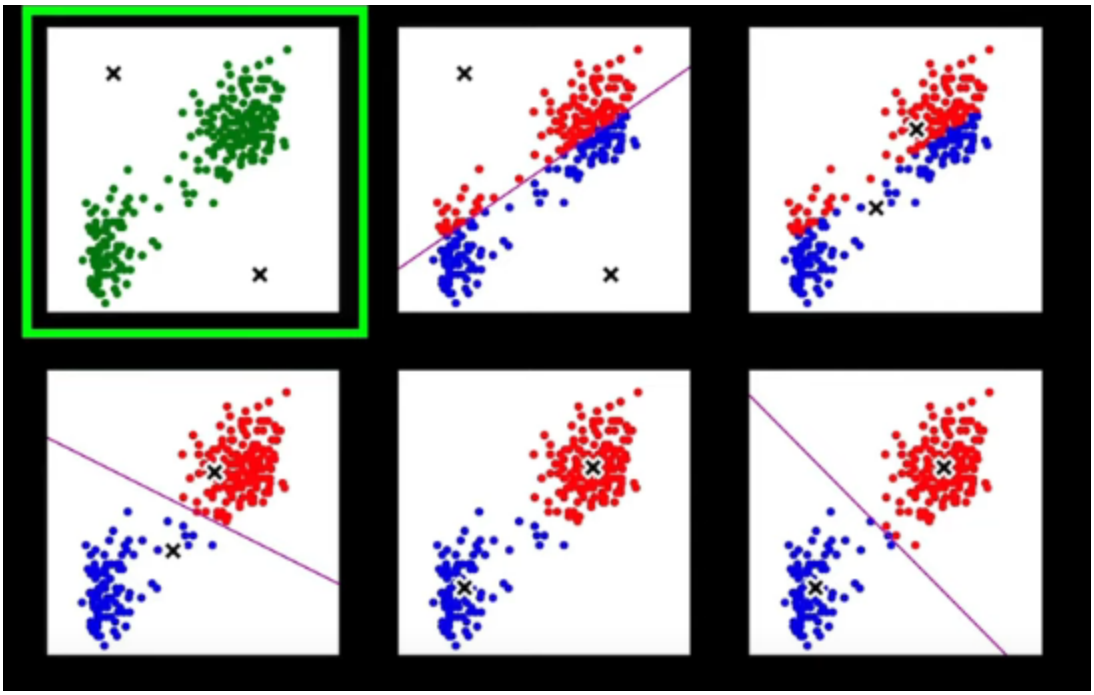
-

hỌC KHÔNG GIÁM SÁT

k-MEAN

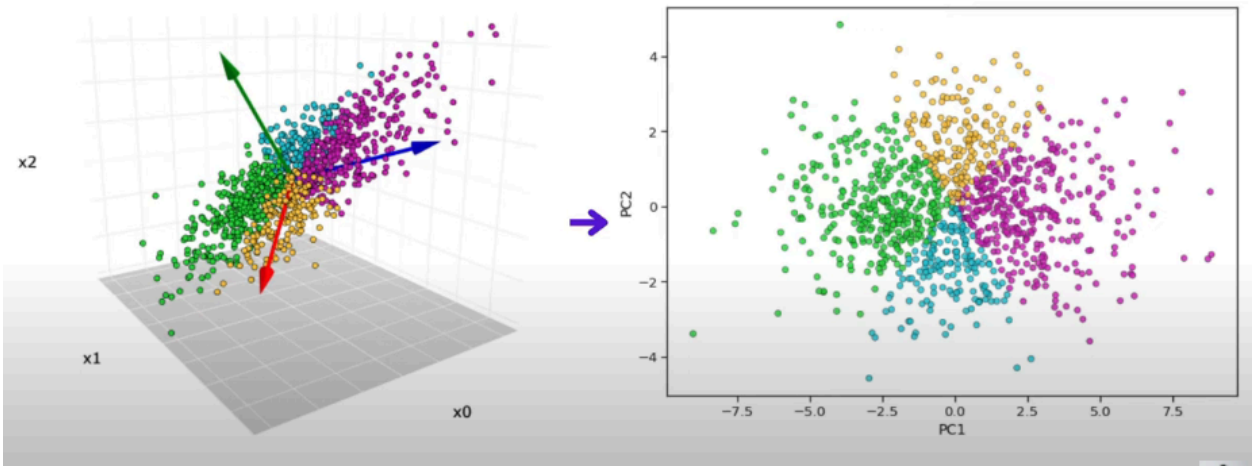
LÀM SAO CHỌN số K tối ưu





PCA

Biến đổi sữ liệu từ N chiều đến 2 chiều để trực qun hóa


x2
x1
x0
PC2
PC1
4
2
0
−2
−4
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5