Họ và tên: Mai Minh Quân

MSSV: 20225661

# K Nearest Neighbors

**Source:**

* KNN Algorithm In Machine Learning - Simplilearn
* K-nearest neighbors - StatQuest

1. Thuật toán KNN

KNN - K Nearest Neighbors, là một trong những thuật toán Học máy có giám sát đơn giản nhất.

Nó phân loại một điểm dữ liệu dựa trên cách các láng giềng của nó được phân loại.

K trong KNN là một tham số đề cập đến số lượng láng giềng gần nhất được đưa vào quá trình đánh giá để phân loại.

A diagram of a diagram of a wine diagram

AI-generated content may be incorrect.

1. Cách lựa chọn giá trị "K"

Không có cách thức nào để xác định giá trị tốt nhất cho 'K', vì vậy có thể phải thử nghiệm một vài giá trị trước khi chọn một giá trị phù hợp

* Các giá trị K nhỏ (như K=1 hoặc K=2) có thể bị nhiễu và chịu ảnh hưởng của các điểm ngoại lai.
* Các giá trị K lớn giảm tác động của nhiễu và ngoại lai, nhưng nếu "K" quá lớn, các danh mục (category) có ít mẫu có thể bị áp đảo bởi các danh mục có nhiều mẫu hơn.

Các gợi ý để lựa chọn giá trị "K":

* Căn bậc hai của "n" với n là tổng số điểm dữ liệu trong tập dữ liệu.
* Chọn "K" là số lẻ giúp tránh nhầm lẫn khi phân loại giữa hai lớp dữ liệu.

1. Khi nào chúng ta sử dụng thuật toán KNN?

Thuật toán KNN được ứng dụng trong các trường hợp sau:

* Dữ liệu được gán nhãn.
* Dữ liệu không có nhiễu.
* Tập dữ liệu có kích thước nhỏ

Lý do là vì KNN không học một hàm phân biệt từ tập dữ liệu huấn luyện.

1. Các bước của thuật toán K-Nearest Neighbors (KNN)

Thuật toán KNN bao gồm 5 bước sau:

* Giả sử D là tập hợp các điểm dữ liệu đã được gán nhãn và A là điểm dữ liệu chưa được gán nhãn cần phân loại.
* Tính toán khoảng cách từ điểm dữ liệu mới A đến tất cả các điểm dữ liệu đã được phân loại trong D.
* Chọn K (một tham số được xác định trước) điểm dữ liệu có khoảng cách nhỏ nhất.
* Kiểm tra nhãn lớp của K láng giềng gần nhất và đếm số lần xuất hiện của mỗi lớp.
* Gán lớp xuất hiện nhiều nhất trong số K láng giềng..

# Linear Regression

**Source:**

* Linear Regression in Python - Machine Learning From Scratch - Patrick Loeber
* Linear Regression – StatQuest

1. Giới thiệu về Linear Regression

Linear Regression là một thuật toán học máy có giám sát được sử dụng để dự đoán các giá trị liên tục. Khác với phân loại, nơi mà mục tiêu là dự đoán các nhãn lớp rời rạc, Linear Regression hướng đến việc tìm ra mối quan hệ tuyến tính giữa các biến đầu vào và biến đầu ra.

Ví dụ:

* Cho một tập dữ liệu gồm các điểm (x, y) trên mặt phẳng, mục tiêu là tìm một đường thẳng (hàm tuyến tính) xấp xỉ tốt nhất các điểm này.
* Hàm tuyến tính được biểu diễn dưới dạng
* : Giá trị dự đoán.
* : Trọng số (độ dốc của đường thẳng).
* : Biến đầu vào.
* : Bias (giao điểm của đường thẳng với trục y).

A diagram of a graph

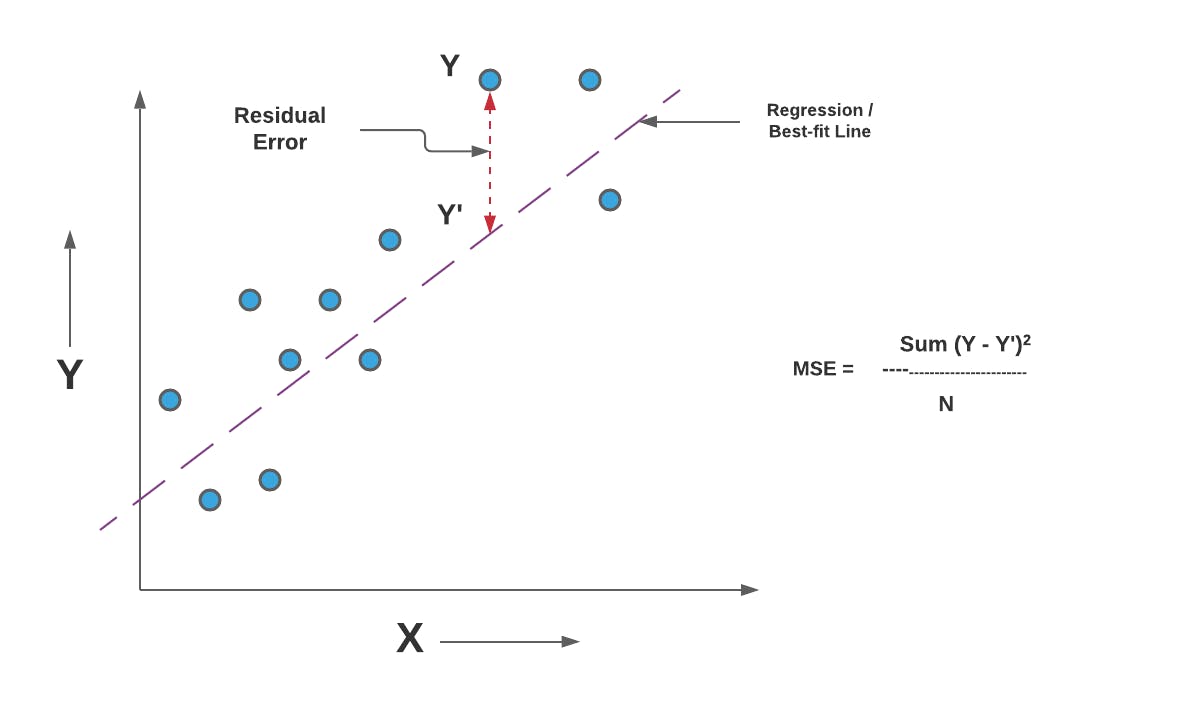
AI-generated content may be incorrect.

1. Hàm chi phí (Cost function)

Mục tiêu của hồi quy tuyến tính là tìm ra các giá trị tối ưu cho và sao cho đường thẳng xấp xỉ dữ liệu một cách chính xác nhất. Để đánh giá mức độ chính xác, chúng ta sử dụng hàm chi phí, thường là Mean Square Error (MSE):

* : Số lượng mẫu dữ liệu.
* : Giá trị thực tế.
* : Giá trị dự đoán.

Mục tiêu là giảm thiểu MSE, tức là tìm và sao cho sai số giữa giá trị dự đoán và giá trị thực tế là nhỏ nhất.



1. Gradient Descend

Để tìm giá trị nhỏ nhất của hàm chi phí, chúng ta sử dụng phương pháp gradient descent. Đây là một phương pháp lặp để tối ưu hóa các tham số của mô hình.

Tính toán Gradient: Tính đạo hàm của hàm chi phí theo () và theo ().

Cập nhật Tham số:

* Cập nhật và theo hướng ngược lại của gradient để giảm thiểu chi phí.
* Công thức cập nhật:
* : Tốc độ học (learning rate), kiểm soát kích thước bước cập nhật.

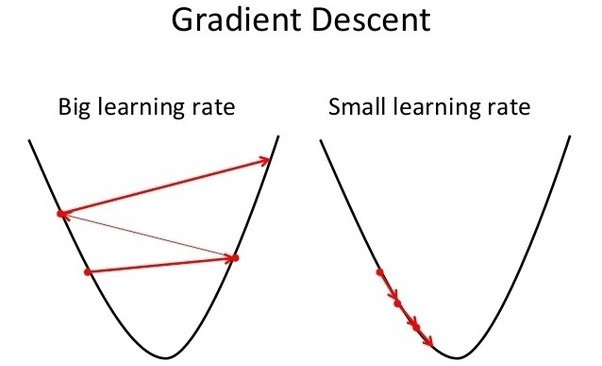
A diagram of a weight loss

AI-generated content may be incorrect.

1. Learning rate

Tốc độ học là một tham số quan trọng ảnh hưởng đến hiệu suất của mô hình.

* Tốc độ học nhỏ: Quá trình huấn luyện diễn ra chậm, nhưng có thể tìm được giá trị tối ưu.
* Tốc độ học lớn: Quá trình huấn luyện diễn ra nhanh, nhưng có thể không ổn định, dẫn đến việc bỏ qua giá trị tối ưu.
* Cần lựa chọn tốc độ học phù hợp để cân bằng giữa tốc độ huấn luyện và độ chính xác của mô hình.



# Logistic Regression

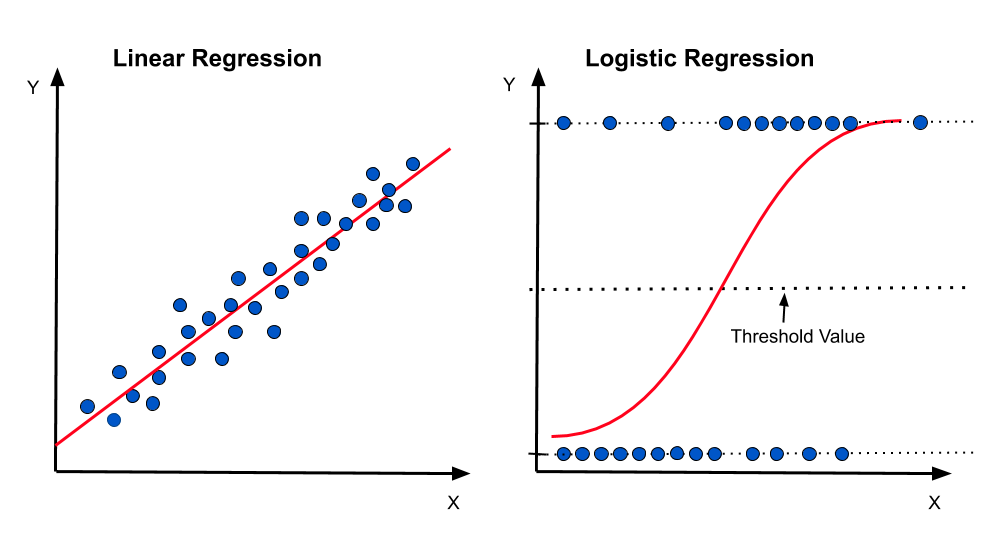
**Source:**

* Logistic Regression in Python - Machine Learning From Scratch - Patrick Loeber
* Logistic Regression – StatQuest

1. Giới thiệu về Logistic Regression

Hồi quy logistic là một thuật toán học máy có giám sát được sử dụng cho các bài toán phân loại nhị phân. Mặc dù có tên gọi là "hồi quy", thuật toán này thực chất được dùng để dự đoán xác suất một mẫu dữ liệu thuộc về một trong hai lớp.

* : Giá trị dự đoán.
* : Trọng số (độ dốc của đường thẳng).
* : Biến đầu vào.
* : Bias.

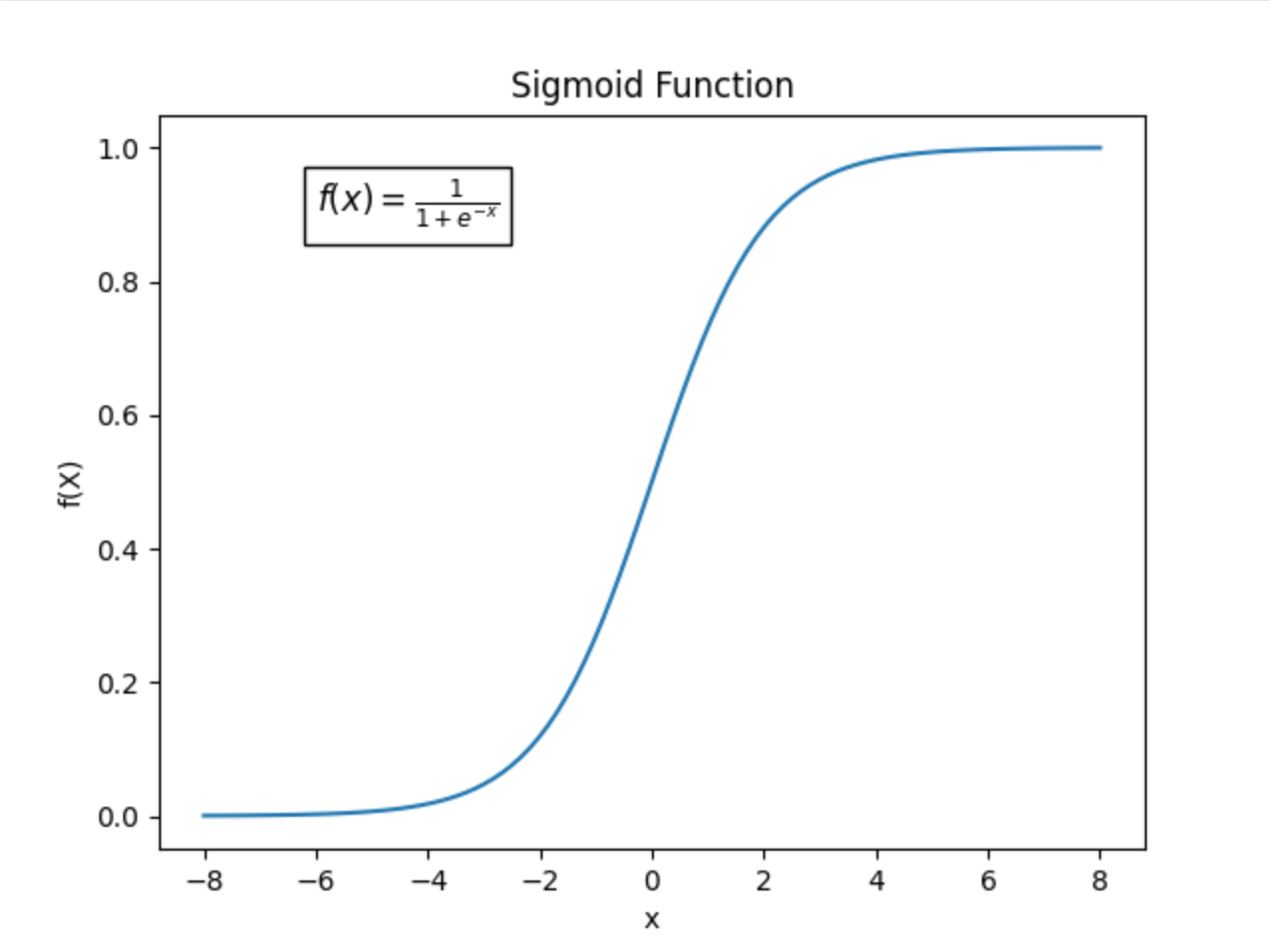


1. Hàm Sigmoid

Khác với hồi quy tuyến tính, dự đoán các giá trị liên tục, hồi quy logistic dự đoán xác suất, nằm trong khoảng từ 0 đến 1.

Để chuyển đổi kết quả đầu ra của mô hình tuyến tính thành xác suất, chúng ta sử dụng hàm sigmoid:

Hàm sigmoid tạo ra một đường cong hình chữ S, giới hạn kết quả đầu ra trong khoảng từ 0 đến 1, biểu diễn xác suất.



1. Loss function

Để tìm các giá trị tối ưu cho và , chúng ta sử dụng hàm chi phí cross-entropy.

Hàm chi phí cross-entropy đo lường sự khác biệt giữa xác suất dự đoán và nhãn lớp thực tế.

Mục tiêu là giảm thiểu hàm chi phí, tìm W và B sao cho xác suất dự đoán gần với nhãn lớp thực tế nhất.

Phương pháp gradient descent được sử dụng để tối ưu hóa các tham số:

* Tính toán gradient (đạo hàm) của hàm chi phí theo và .
* Cập nhật và theo hướng ngược lại của gradient để giảm thiểu chi phí.
* Tốc độ học (learning rate) là tham số quan trọng, kiểm soát kích thước bước cập nhật.
* Quy tắc cập nhật

: Tốc độ học (learning rate)