**GIỚI THIỆU VỀ CÁC THƯ VIỆN PYTHON:**

1. Thư viện numpy (Numeric Python)

* Thư viên numpy của ngôn ngữ Python là 1 thư viện cốt lõi, hữu ích trong các tính toán khoa học. Nó cung cấp đối tượng mảng đa chiều hiệu suất cao cùng với các công cụ, các hàm và thuộc tính để làm việc với chúng.
* Numpy cung cấp cho ta khả năng tạo đối tượng mảng N chiều, giúp ta biểu diễn các mô hình toán học như ma trận, vecto, tensor,.. cũng như các hàm toán học thực hiện phép toán trên mảng.
* Trong AI/ML:
* Các dữ liệu đầu vào của mạng neural thường được biểu diễn dưới dạng 1 tensor nhiều chiều
* VD: Một hình ảnh RGB có thể được biểu diễn dưới dạng một tensor 3 chiều (chiều cao, chiều rộng, 3 kênh màu)
* Các trọng số (weights) và bias của các neuron trong mạng neural cũng được biểu diễn dưới dạng các tensor. Chúng lưu trữ thông tin về cách các neural kết nối với nhau và cách chúng xử lý dữ liệu. ( Trong hidden layer , các đầu ra từ neural của các lớp trước tạo thành 1 tensor, chập với filter là 1 ma trận các weight sau đó cộng với bias và thế vào activation fuunction để cho ra KQ).
* Một số các hàm và thuộc tính của thư viện numpy:

|  |  |
| --- | --- |
| np.array(list) | Tạo một mảng từ một danh sách |
| np.zeros((shape)) | Tạo một mảng đầy các số 0 với hình dạng cho trước |
| np.ones((shape)) | Tạo một mảng đầy các số 1 với hình dạng cho trước |
| shape | Trả về hình dạng của mảng |
| reshape(newshape) | Thay đổi hình dạng của mảng |
| Mean() | Giá trị trung bình |
| Std() | Độ lệch chuẩn |

1. Thư viện Matplotlib

* Thư viện Matplotlib là thư viện vẽ đồ thị của Python. Nó cung cấp 1 loạt các công cụ để tạo ra các biểu đồ dạng khác nhau giúp ta trực quan hóa dữ liệu một cách hiệu quả. Đồng thời, thư viện còn cung cấp cho ta các hàm cần thiết đẻ điều chỉnh các thuộc tính của biểu đồ ( màu sắc, nhãn trục và bố cục tổng thể,,...)
* Thư viện Matplotlib đóng vai trò vô cùng quan trọng trong lĩnh vực AI/ML:
* Hiển thị phân bố dữ liệu: Phân bố dữ liệu dựa trên các thuộc tính của chúng 🡪 thấy được các nhóm dữ liệu có cùng đặc trưng, các điểm bất thường hoặc xu hướng tương lai của chúng.
* Phát hiện các MQH giữa 2/nhiều hệ dữ liệu khác nhau, từ đó xây dựng các mô hình dự đoán
* So sánh, đánh giá hiệu quả về mặt hiệu suất, độ chính xác của mô hình một cách tương đối so với thực tế.
* Một số loại biểu đồ và thuộc tính

|  |  |
| --- | --- |
| plt.hist() | Vẽ biểu đồ phân bố dữ liệu theo tần suất |
| plt.scatter() | Vẽ biểu đồ phân tán rời rạc trên KG 2 chiều |
| plt.pie() | Vẽ biểu đồ tròn phân bố dữ liệu theo tỷ lệ |
| plt.bar() | Vẽ biểu đồ cột của dữ liệu |
| plt.plot() | Vẽ biểu đồ đường dữ liệu liên tục |
| Plt.x/ylabel | Dán nhãn cho trục x/y của đồ thị |
| Plt.legend() | Tự động phân biệt 2 đối tượng khác biệt (cột, đường, nhóm điểm ) trên biểu đồ |

1. Thư viên Pandas

* Thư viện Pandas là 1 thư viện mã nguồn mở của ngông ngữ Python được dùng với mục đích xử lý dữ liệu 1 cách hiệu quả, giúp ta thao tác, phân tích và trích xuất thông tin từ các cơ sở dữ liệu dễ dàng.
* Thư viện Pandas bao gồm 2 đối tượng chính:
* Series: mảng 1 chiều với thuộc tính và các giá trị ( đóng vai trò tương tự như các cột trong bảng )
* DataFrame : Một bảng dữ liệu 2 chiều, gồm nhiều Series (Cột ) và hàng (bản ghi ) khác nhau. Đây là đối tượng và loại cấu trúc dữ liệu chính được sử dụng phổ biến nhất trong thư viện này
* Thư viện Pandas đóng vai trò quan trọng trong việc chuẩn bị dữ liệu/ xử lý dữ liệu ban đầu trong các mô hình học máy và học sâu:
* Về việc chuẩn bị dữ liệu: Thư viện hỗ trợ đọc dữ liệu từ nhiều nguồn, nhiều định dạng khác nhau ( .csv,.exl,.sql,.....) để hiển thị ban đầu.
* Tiền xử lý dữ liệu: chuẩn hóa các vấn đề về dữ liệu (bổ sung các trường thiếu, dữ liệu trùng lặp, dữ liệu sai định dạng ,...) hay có thể tạo các đặc trưng mới có ý nghĩa để cải thiện thêm hiệu suất mô hình.
* Kết hợp các thư viện như Matplotlib để trực quan hóa dữ liệu, từ đó tìm ra MQH giữa các thuộc tính, các biến của dữ liệu

**TÓM TẮT VỀ LINEAR REGRESSION VÀ GRADIENT DESCENT**

* Linear regression model (hồi quy tuyến tính ) là mô hình dùng để mô hình hóa mối quan hệ tuyến tính của các thuộc tính của sự vật, từ đó cho ta biết được dự đoán xu hướng của thuộc tính cần tìm dựa trên các thuộc tính đầu vào.
* Ví dụ: dựa đoán cân nặng dựa trên chiều cao, cân nặng là biến phụ thuộc tuyến tính với biến chiều cao.

Dạng toán học tổng quát: y=a1x1+a2x2+a3x3+...+aNxN+ b :  
a1,a2,... là các trọng số weight tương ứng từng biến độc lập

B: hệ số tự do được quy định như giá trị ban đầu của y

* Quy trình xây dựng hệ thống mô hình hồi quy tuyến tính:

A diagram of a learning algorithm

Description automatically generated

* Chuẩn bị bộ dữ liệu mẫu : dữ liệu đã được xác định đầu ra chính xác
* Xây dựng mô hình và thuật toán học
* Cho bộ dữ liệu mẫu vào trong mô hình để training cho đến khi đạt được hàm loss function min (hiệu chỉnh các hệ số/ trọng số weight của ma trận hệ số)
* Khi ta đưa đầu vào mới, đầu ra của mô hình sẽ là dự đoán xấp xỉ của giá trị cần tìm dựa trên tính tuyến tính của mô hình.
* Một số các thông số cơ bản:
* Cost Function(Sai số dự đoán): nửa bình phương hiệu giữa giá trị thực và giá trị dự đoán khi training 🡪 đóng vai trò quan trọng trong việc tiền xử lý dữ liệu ngoại lai và train mô hình để tìm hệ số tối ưu.
* Loss function (hàm mất mát): bằng tổng các sai số của tất cả các dữ liệu. Hàm này càng nhỏ thì chứng tỏ đường thẳng của mô hình đi qua gần các điểm. Đến đây, yêu cầu đặt ra là bài toán tìm giá trị nhỏ nhất của hàm mất mát để tối ưu trọng số
* Gradient descent: ý tưởng của thuật toán là tìm giá trị nhỏ nhất của hàm mất mát bằng cách thay đổi các giá trị trọng số theo hướng giảm gradient đến khi về 0🡪 Điểm cực tiểu ở đó hàm mất mát đạt GTNN
* Thuật toán:

A diagram of a function

Description automatically generated

* Tìm vi phân của hàm mất mát theo các trọng số /hệ số tự do ( dJ/dw)
* Trọng số mới được tính:

New\_w=w-a\*(dJ/dw) với a là hệ số học nằm trong khoảng từ 0 đến 1

* Tiếp tục cho đến khi vi phân tiến đến gần hoặc xấp xỉ 0 🡪 thuật toán kết thúc và ta có được các giá trị hệ số tối ưu cần tìm cho mô hình
* Càng gần giá trị min của hàm, vi phân càng bé nên sự thay đổi của hàm càng bé 🡪 việc chọn hệ số học a có vai trò quan trọng quyết định đến thời gian và kết quả của qáu trình.
* Ứng dụng trong bài toán hồi quy tuyến tính: Do hàm mất mát của mô hình này có bình phương mà có tuyến tính theo x 🡪 ta chỉ có một min duy nhất:

J=...(wx+b-y)^2🡪 dJ/dw=2(wx+b-y)\*x 🡪 hàm bậc 2 có duy nhất 1 cực trị

**TÓM TẮT VỀ LOGISTIC REGRESSION**

* Hồi quy logistic là mô hình sử dụng trong học máy để dự đoán kết quả thuộc về 1 trong 2/ nhiều nhóm phân loại. Nó giúp chúng ta trả lời câu hỏi rằng dữ liệu đầu vào mang tính đúng/sai, có/không hay thuộc 1 nhóm đặc trưng nào đó với xác suất là bao nhiêu
* Những điểm khác biệt so với hồi quy tuyến tính
* Giá trị đầu ra nằm trong khoảng từ [0,1] (Xác suất )
* Mối quan hệ giữa biến độc lập và biến phụ thuộc không còn là qua hệ tuyến tính mà trở nên phức tạp hơn
* Tập dữ liệu mẫu không còn là cặp (dữ liệu vào, kết quả ra ) mà là dữ liệu được dán nhãn /phân loại sẵn
* Do MQH ko còn tuyến tính và tập đầu ra mới 🡪 sử dụng hàm sigmoid làm hàm kích hoạt để thỏa mãn yêu cầu bài toán. Hàm này cũng có đạo hàm nên dễ dàng áp dụng thuật toán gradient descent để tối ưu hóa trọng số mô hình .
* Các bước xây dựng mô hình logistic regression đơn giản:
* Xây dựng mô hình và chuẩn bị dữ liệu mẫu đã dán nhãn
* Thiết lập loss function và tối ưu hóa trọng số bằng các thuật toán
* Đưa ra dự đoán dữ liệu mới bằng model vừa tìm được. Cần đưa ra ranh giới quyết định để xác định rõ ràng xác suất của dữ liệu thuộc nhóm xác định là bao nhiêu (tương ứng với độ chính xác )
* Hàm loss function của mô hình này :

J=−(*yi*​∗*log*(*yi*​^​)+(1−*yi*​)∗*log*(1−*yi*​^​))

* Nếu yi=1🡪 J= -log(yi). Nếu mô hình dự đoán 1 thì độ sai khác giữa dự đoán và thực tế là 0 và ngược lại nếu giá trị dự đoán là 0 thì sự sai khác là rất lớn
* Tương tự với trường hợp yi=0
* Quay trở về bài toán tìm các hệ số w để hàm mất mát là nhỏ nhất
* Ứng dụng : Áp dụng mô hình cho bài toán phân loại dữ liệu: bằng cách so sánh 2 biểu thức xác suất dữ liệu thuộc nhóm nào cao hơn

Mục tiêu tuần tới: Phân tích kĩ càng VD về linear regression/ logistic regression , làm BT và tìm hiểu về K-means Clustering and Applications !