Logistic Regression, K – Nearest Neightbors, Decision Tree

1. Tiền xử lý dữ liệu

Các cột nên loại bỏ:

|  |  |
| --- | --- |
| Id (Mã định danh) | Không có ý nghĩa phân tích |
| Age (Tuổi) | Tất cả là học sinh |
| City ( Thành phố) | Rất nhiều giá trị, không cần thiết nếu không phân tích địa phương |
| Profession (nghề nghiệp) | Gần như tất cả là "Student" |
| Work Pressure(Áp lực công việc) | Tất cả giá trị là 0 |
| Job Satisfaction ( Công việc) | Hầu hết là 0 |
| Degree ( Bằng cấp) | Có nhiều giá trị khác nhau, không quan trọng với mục tiêu hiện tại |

Các cột giữ lại và tiền xử lý dữ liệu

Scale( Tạo tỉ lệ)

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Ý Nghĩa | Loại dữ liệu | Cách xử lý cần thiết |
| Gender | Giới tính | Object | Chuyển thành số: Male = 0, Female = 1 |
| Academic Pressure | Mức độ áp lực học tập | Float | Scale (StandardScaler) |
| CGPA | Điểm trung bình tích lũy | Float | Scale |
| Study Satisfaction | Mức độ hài lòng với việc học | Float | Scale |
| Sleep Duration | Thời lượng ngủ | Object | Map thành số: ví dụ Less than 5 hours → 4.5 |
| Dietary Habits | Thói quen ăn uống | Object | Chuyển thành số: Healthy = 0, Moderate = 1, Unhealthy = 2 |
| Have you ever had suicidal thoughts? | Đã từng có suy nghĩ tự tử? | Object | Yes = 1, No = 0 |
| Work/Study Hours | Số giờ học/làm mỗi ngày | Float | Scale |
| Financial Stress | Mức độ căng thẳng tài chính, tiền bạc | Float | Scale |
| Family History of Mental Illness | Có người thân từng bị bệnh về tâm thần không? | object | Yes = 1, No = 0 |
| Depression | Kết quả Trầm Cảm | int | Nhãn (label) – giữ nguyên |

1. Logistic Regression (Hồi quy logistic)

Hồi quy logistic là mô hình dùng để dự đoán xác suất một sự kiện nào đó xảy ra, ví dụ như: "Người này có bị bệnh hay không?" hay "Email này có phải là spam không?". Mô hình này hoạt động bằng cách tính toán một tổ hợp tuyến tính của các đặc trưng đầu vào (như tuổi, giới tính, thu nhập…) và sau đó biến kết quả đó thành một giá trị xác suất trong khoảng từ 0 đến 1 thông qua hàm sigmoid.

Công thức tính toán trong mô hình logistic là:

​ f(x) =

Trong đó :

Trong hồi quy logistic, trước khi đưa ra dự đoán, mô hình sẽ thực hiện một bước gọi là tính tổ hợp tuyến tính, ký hiệu là x. Đây là tổng của các đặc trưng đầu vào (như tuổi, giới tính, điểm số, v.v...) sau khi đã nhân với hệ số phù hợp, rồi cộng thêm một số điều chỉnh (gọi là bias).

Công thức :

𝑦̂ = 𝑤1𝑥1 + 𝑤2𝑥2 + . . . 𝑤𝑘𝑥𝑘 + b

Trong đó :

* x1, x2,…., xn: là các đặc trưng (thông tin đầu vào).
* w1, w2,….., wn: là các hệ số (trọng số) mà mô hình học được.
* b: là một số điều chỉnh (bias), giúp mô hình linh hoạt hơn.

2. Lý thuyết các mô hình học máy

2.1 Mô hình Logistic Regression

2.1.1 Giới thiệu về thuật toán

Hồi quy logistic là một phương pháp thống kê phổ biến để phân loại nhị phân. Nó được sử dụng rộng rãi trong nhiều lĩnh vực như y học, kinh tế, sinh học và khoa học xã hội để dự đoán khả năng xảy ra của một sự kiện. Mô hình hồi quy logistic dựa trên việc sử dụng hàm logistic (hàm sigmoid) để biến đổi kết quả tuyến tính thành xác suất. giúp dự đoán các kết quả nhị phân.

Ý tưởng cơ bản của hồi quy logistic là thay vì dự đoán trực tiếp giá trị của biến phụ thuộc (thường là 0 hoặc 1), mô hình sẽ dự đoán xác suất của sự kiện đó xảy ra. Sau đó, xác suất này có thể được sử dụng để đưa ra quyết định phân loại.

Hàm sigmoid được định nghĩa như sau:

Hàm này khi được biểu diễn dưới dạng đồ thị có dạng chữ S, các giá trị của x đều đưa ra kết quả f(x) có giá trị từ 0 đến 1. Điều này rất phù hợp với việc mô hình hóa xác suất.

Với bài toán hồi quy tuyến tính, kết quả của biến phụ thuộc y được đưa ra bởi công thức:

𝑦̂ =

Trong đó:

* x là tổ hợp tuyến tính của các đặc trưng đầu vào.
* 𝑦̂: biến phụ thuộc cần dự đoán.
* 𝑥1, 𝑥1, . . ., 𝑥𝑘 là các biến độc lập (biến đặc trưng).
* 𝑤1, 𝑤2, . . ., 𝑤𝑘 là các hệ số hồi quy.
* b là sai số.

Công thức trên có thể viết gọn lại bằng vector thành:

𝑦̂ = 𝑤𝑇. 𝑥 + 𝑏

Trong hồi quy logistic, 𝑦̂ sẽ được đưa qua hàm sigmoid để tính được xác suất xảy ra.

2.1.2 Xây dựng thuật toán Logistic Regresison  
 Logistic regression là một thuật toán học máy có giám sát với nhiệm vụ phân loại nhị phân bằng cách dự đoán xác suất của một kết quả, sự kiện hoặc quan sát. Mô hình đưa ra kết quả nhị phân hoặc phân đôi được giới hạn ở hai kết quả có thể xảy ra : có – không, 0 – 1 hoặc đúng – sai

Logistic regression sử dụng hàm logistic được gọi là hàm sigmoid để ánh xạ các dự đoán

và xác suất của chúng. Hàm sigmoid đề cập đến một đường cong chữ S chuyển đổi bất

kỳ giá trị thực nào thành phạm vi từ 0 đến 1.

Hàm sigmoid:

Trong đó:

* f(x) là xác xuất đầu ra (có giá trị giữa 0 và 1)
* x = 𝑤1𝑥1 + 𝑤2𝑥2 + . . . + 𝑤𝑛𝑥𝑛 + 𝑏 là một tổ hợp tuyến tính của các biến đầu vào

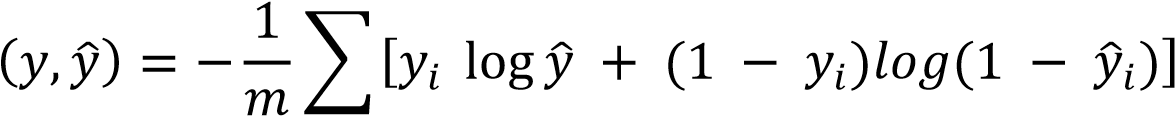
𝑥1, 𝑥2, 𝑥𝑛 với các trọng số tương ứng 𝑤1, 𝑤2, 𝑤𝑛 và bias(hệ số điều chỉnh) b.

Công thức cross-entropy của hồi quy logistic được cho bởi:

𝐿(𝑦, 𝑦̂) = − [𝑦 ∗ log 𝑦̂ + (1 − 𝑦) ∗ log (1 − 𝑦̂)]

Hàm mất mát (loss function) trong logistic regression là hàm cross-entropy. Hàm này đo lường sự khác biệt giữa giá trị dự đoán của mô hình và giá trị thực tế. Công thức của hàm mất mát trong bài toán này là này là:

m



𝑖 = 1

Trong đó:

• M là số lượng mẫu trong tập dữ liệu

• 𝑦𝑖 là giá trị thực tế của nhãn (0 hoặc 1)

• 𝑦̂𝑖 là giá trị dự đoán (xác xuất) của mô hình

Để thuật toán đạt độ chính xác cao nhất, việc của chúng ta là tối ưu hàm mất mát này sao cho giá trị của hàm là nhỏ nhất. Một cách phổ biến có thể nhắc tới là Gradient Descent.

Quá trình tối ưu hóa: để tìm ra các trọng số tối ưu w và bias b, ta sử dụng phương pháp tối ưu hóa như Gradient Decesnt. Quá trình này bao gồm các bước sau:

1. Khởi tạo các trọng số và bias ngẫu nhiên hoặc bằng 0
2. Tính toán giá trị dự đoán 𝑦̂ cho mỗi mẫu dữ liệu sử dụng hàm sigmoid.
3. Tính toán hàm mất mát sử dụng công thức cross-entropy.
4. Cập nhật trọng số và bias bằng cách giảm thiểu hàm mất mát thông qua Gradient Descent, Quá trình này lặp lại cho đến khi hàm mất mát hội tụ đến giá trị tối thiểu hoặc đạt đến số lượng vòng lặp tối đa.

Một số nhược điểm của logistic regression:

• Dễ bị ảnh hưởng bởi các điểm dữ liệu ngoại lai

• Yêu cầu chuẩn bị dữ liệu cẩn thận

• Không xử lý tốt dữ liệu mất cân bằng

**K-Nearest Neighbors và Decision Tree không có công thức tổng quát cụ thể giống như Logistic Regression , cả 2 mô hình này hoạt động theo thuật toán, không dựa trên các biểu thức toán học.**

2.2 Mô hình K – Nearest Neightbors

Được xây dựng dựa trên công thức tỉnh khoảng cách:



* So sánh điểm dữ liệu mới với tất cả các điểm đã biết.
* Chọn K điểm gần nhất, rồi dùng đa số để phân loại

Thuật toán KNN là một trong những phương pháp học máy có giám sát (supervised learing) không tham số, nghĩa là nó không đưa ra giả định nào về dữ liệu. Phương pháp này đơn giản nhưng vô cùng hiệu quả, nó được ứng dụng rộng rãi trong nhiều lĩnh vực như phân loại, dự đoán và khai phá dữ liệu.

Ý tưởng cốt lõi của KNN dựa trên nguyên lý “những vật thể gần nhau thường có xu hướng tương đồng”. Khi cần phân loại một điểm dữ liệu mới, KNN sẽ xem xét K điểm dữ liệu gần nhất trong tập huấn luyện và dựa vào đó để đưa ra quyết định. Nếu K điểm dữ liệu đó đa số thuộc về lớp A thì điểm dữ liệu mới cũng sẽ được gán vào lớp A.

Ưu điểm lớn nhất của KNN là đơn giản và dễ hiểu. Thuật toán không yêu cầu xây dựng mô hình phức tạp hay ước lượng tham số, mọi tính toán đều được thực hiện trực tiếp khi cần dự đoán. Điều này giúp KNN trở thành lựa chọn tốt cho những bài toán có dữ liệu nhỏ hoặc yêu cầu thời gian xử lý nhanh.

2.2.1 Xây dựng công thức

Tính khoảng cách: Chọn một độ đo khoảng cách như Euclidean, Manhattan, Minkowski,…

* Chọn K láng giềng gần nhất: Sau khi sắp xếp các điểm trong tập dữ liệu theo thứ tự tăng dần của khoảng cách đã tính, thuật toán chọn ra K điểm dữ liệu gần nhất với điểm cần xét.
* Phân loại: Đếm số lượng các điểm dữ liệu thuộc mỗi lớp trong K láng giếng đó, gán nhãn của điểm dữ liệu cần dự đoán cho lớp có số lượng điểm dữ liệu là nhiều nhất.

Giả sử điểm x (𝑥1, 𝑥2, . . . , 𝑥𝑛) và y (𝑦1, 𝑦2, . . . , 𝑦𝑛), chúng ta có thể áp dụng các công

thức tính khoảng cách thường sau tùy thuộc vào từng yêu cầu của bài toán:

* Euclidean:

A mathematical equation with a square and square with a square and square with a square and a square with a square and a square with a square and a square with a square and a square with a

AI-generated content may be incorrect.

* Manhattan

* Minkowski:

A black and white math symbol

AI-generated content may be incorrect.

A screenshot of a game

AI-generated content may be incorrect.

Minh họa về thuật toán KNN.

Tại ví dụ trong ảnh, với K = 9, do số điểm màu đỏ là 7 và số điểm màu xanh là 2 nên điểm được dự đoán sẽ được gấn cho lớp màu đỏ.

Một số nhược điểm của thuật toán KNN:

* Nhạy cảm với nhiễu
* Tốn kém về tính toán
* Hiệu suất giảm khi số chiều tăng
* Không xử lý tốt dữ liệu mất cân bằng

2.3 Mô hình cây quyết định (Decision Tree)

• Tại mỗi bước, chọn thuộc tính và điều kiện chia tách tốt nhất để phân loại dữ liệu.

• Xây cây bằng cách đặt các điều kiện dạng "if - then"

Cây quyết định (Decision Tree) là một trong những kỹ thuật học máy phổ biến được sử dụng trong cả phân loại và hồi quy. Một cây quyết định mô phòng quá trình đưa ra quyết định của con người bằng cách chia dữ liệu thành các nhóm con dựa trên các thuộc tính, từ đó xây dựng một cấu trúc dạng cây với các nút quyết định và lá.

Ý tưởng chính của cây quyết định là xây dựng một mô hình dạng cây để dự đoán giá trị của các biến mục tiêu dựa trên các thuộc tính của dữ liệu đầu vào. Mỗi nút trong cây đại diện cho một thuộc tính của dữ liệu, mỗi nhánh đại diện cho một giá trị của thuộc tính đó, và mỗi lá đại diện cho một giá trị dự đoán của biến mục tiêu. Cây quyết định phân chia dữ liệu dựa trên các thuộc tính sao cho sự không đồng nhất của dữ liệu trong nhóm con là thấp nhất có thể.

Cây quyết định là một phương pháp thông dụng trong khai phá dữ liệu. Nó quyết định mô tả cấu trúc của một cây, trong đó các lá đại diện cho các phân loại còn cành đại diện cho các kết hợp của các thuộc tính dẫn tới phân loại đó. Một quyết định có thể được học bằng cách chia tập hợp nguồn thành các tập con dựa theo một kiểm tra giá trị thuộc tính. Quá trình kiểm tra này được lặp lại đệ quy cho mỗi tập con dẫn xuất.

Thuật toán cây quyết định là một trong những thuật toán học máy phổ biến và dễ hiểu nhất. Nó được sử dụng cho cả bài toán phân loại về hồi quy. Mô hình cây quyết định là một cấu trúc phân cấp gồm các nút và nhánh. Trong đó bao gồm:

* Nút gốc (root node): Đại diện cho toàn bộ tập dữ liệu .
* Nút quyết định (decision node): Đại diện cho một câu hỏi hoặc điều kiện kiểm tra trên một thuộc tính .
* Nhánh (branch): Đại diện cho kết quả của câu hỏi hoặc điều kiện kiểm tra.
* Nút lá (leaf node): Đại diện cho lớp hoặc giá trị dự đoán.

Thuật toán bắt đầu bằng cách chọn thuộc tính có khả năng phân chia tập dữ liệu tốt nhất dựa trên một tiêu chí đo lường như entropy, information gain hoặc gini index. Sau đó chia tập dữ liệu thành các tập con dựa trên giá trị của thuộc tính đã chọn cho đến khi tất cả các nút lá đều thuần nhất (chỉ chứa một lớp) hoặc đạt đến một điều kiện dừng (ví dụ: độ sâu tối đa của cây).

Tóm tắt sự khác biệt giữa các mô hình:

• Logistic Regression: Tính toán xác suất xảy ra của sự kiện dựa trên các đặc trưng đầu vào, kết quả cho ra sẽ là 1 xác suất.

• KNN: Dự đoán dựa trên sự so sánh với những điểm dữ liệu gần nhất.

• Decision Tree: Dự đoán qua một chuỗi các câu hỏi điều kiện, phân chia dữ liệu thành các nhánh để ra quyết định.