# Mô hình hóa dữ liệu

## Chia bộ dữ liệu train/test

Bộ dữ liệu sau khi xử lý (sau đây gọi bộ dữ liệu ban đầu) được chia thành 2 bộ dữ liệu train và bộ dữ liệu test theo tỷ lệ tương ứng là 8:2. Chi tiết kích thước các bộ dự liệu được tính theo bảng 1.

|  |  |  |  |
| --- | --- | --- | --- |
| Tỷ lệ train/test = 8/2 | | | |
|  | Bộ dữ liệu ban đầu | Bộ dữ liệu train | Bộ dữ liệu test |
| Kích thước (bản ghi) | 5358 | 4286 | 1072 |

Bảng . Kích thước các bộ dữ liệu.

## Các mô hình sử dụng

1. Cây quyết định (Decision tree)
2. Cơ sở lý thuyết:

Thuật toán tạo ra mô hình học có cấu trúc dạng cây, mỗi nút trong cây nếu là nút nhánh sẽ chứa giá trị ngưỡng và chỉ số thuộc tính cần so sánh; nếu là nút lá sẽ chứa nhãn hoặc giá trị dự đoán được.

Thuật toán xây dựng cây quyết định có các bước cơ bản như sau [1]:

1. Khởi tạo tập thuộc tính S bằng tập thuộc tính train; khởi tạo cây rỗng T.
2. Lựa chọn thuộc tính tốt nhất nhất (attrib) trong tập S bằng cách đánh giá tập thuộc tính S theo một tiêu chí nhất định và lựa chọn ra thuộc tính có giá trị đánh giá “tốt nhất”. Hai tiêu chí thường dùng để lựa chọn thuộc tính cho nút là Entropy và Information Gain (nếu dự đoán dữ liệu rời rạc) hoặc Gini index (nếu dự đoán dữ liệu liên tục).
3. Tạo nút lá hoặc nút nhánh mới trong cây tùy vào giá trị đánh giá của attrib.
4. Chia nhỏ tập S thành tập thuộc tính S’.
5. Với mỗi nhánh mới của nút, quay lại bước 2 với S=S’ và cây T=nút N.

Để dự đoán, thuật toán thực hiện duyệt cây từ nút gốc. Ở mỗi nút, thuật toán chọn nhánh dựa vào kết quả phép so sánh giữa giá trị ngưỡng lưu trong nút đó với giá trị của thuộc tính ở bộ dữ liệu input tương ứng chỉ số thuộc tính lưu trong nút. Thuật toán trả về giá trị nút lá đầu tiên tìm được.

1. Các tham số dùng để xây dựng mô hình được cài đặt theo hàm API của thư viện sklearn như bảng sau.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên tham số | Giá trị | Mô tả |
| 1 | criterion | 'mse' | Hàm số đánh giá chất lượng của nút nhánh. |
| 2 | splitter | 'best' | Phương pháp phân nhánh một nút. |
| 3 | max\_depth | None | Độ sâu tối đa của cây. |
| 4 | min\_samples\_split | 2 | Số mẫu tối thiểu để tách một nút nhánh. |
| 5 | min\_samples\_leaf | 1 | Số mẫu tối thiểu để trở thành nút lá. |
| 6 | min\_weight\_fraction\_leaf | 0.0 | Phân số có trọng số tối thiểu của tổng các trọng số để giữ lại ở nút lá. |
| 7 | max\_features | None | Số thuộc tính xét đến khi tách nhánh. |
| 8 | random\_state | None | Điều khiển mức độ ngẫu nhiên của mô hình học. |
| 9 | max\_leaf\_nodes | None | Số nút lá tối đa. |
| 10 | min\_impurity\_decrease | 0.0 | Ngưỡng pha tạp để trở thành nút nhánh. |
| 11 | ccp\_alpha | 0.0 | Tham số độ phức tạp dùng cho thuật toán tỉa cây Minimal Cost-Complexity Pruning. |

Bảng . Chi tiết tham số hàm API tạo mô hình cây quyết định.

1. Hồi quy tuyến tính (Linear regression)
2. Cơ sở lý thuyết:

Thuật toán giả định mối quan hệ tuyến tính giữa tập thuộc tính đầu vào (*X ∈ Rn*) và thuộc tính đầu ra (*y ∈ R*) theo công thức:

Khi thực hiện, thuật toán lặp lại việc cập nhật các trọng số (*W ∈ Rn*) và bias (*b ∈ R*) dựa vào giá trị hàm lỗi sao cho hàm chi phí đạt giá trị nhỏ nhất. Thông thường, các trọng số và bias được đặt bằng 0 hoặc được đặt giá trị ngẫu nhiên; hàm lỗi thường được sử dụng như Linear Least Squares, Sigmoid Function, …; hàm cost thường là giá trị trung bình của hàm lỗi cho mỗi mẫu dữ liệu.

1. Các tham số dùng để xây dựng mô hình được cài đặt theo hàm API của thư viện sklearn như bảng 3. Hàm lỗi được sử dụng là hàm Ordinary Least Squares có công thức như sau:

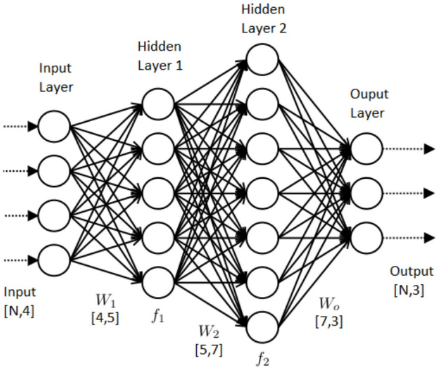
[2]

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên tham số | Giá trị | Mô tả |
| 1 | fit\_intercept | True | Chỉ định có nên tính intercept hay không. |
| 2 | normalize | False | Chỉ định có nên chuẩn hóa dữ liệu trước hay không. |
| 3 | copy\_X | True | Chỉ định có nên copy bộ dữ liệu train hay không. |
| 4 | n\_jobs | None | Số tác vụ thực hiện tính toán. |
| 5 | positive | False | Chỉ định các trọng số đều dương hay không. |

Bảng . Chi tiết tham số hàm API tạo mô hình hồi quy tuyến tính.

1. Mạng nơ-ron nhân tạo nhiều lớp (Multi-layer Perceptron Regression, MLP)
2. Cơ sở lý thuyết:

Mô hình được sử dụng để “học” một hàm số phi tuyến tính , với *m* là số chiều của input và *o* là số chiều của output. Khác với mô hình hồi quy tuyến tính, MLP có một hoặc nhiều hơn các lớp phi tuyến tính (lớp ẩn, hidden layer). Mỗi nơ-ron thứ *i* ở một lớp được nối với mọi nơ-ron thứ *j* ở lớp tiếp theo và được gán các trọng số *wij*; mỗi lớp *l* trong mô hình có *d* nơ-ron thì có *d* bias, .



Hình . Ví dụ về một mạng MLP với số chiều cụ thể.

Mỗi nơ-ron thứ *j* ở lớp ẩn thứ *l* biến đổi outputcủa các nơ-ron ở lớp *(l-1)* trước đó theo công thức:

[3]

Output này sau đó được áp dụng một hàm kích hoạt phi tuyến tính để tính toán ra giá trị output cuối cùng của nơ-ron thứ *i* ở lớp thứ *l*:

Các nơ-ron ở lớp output có thể không có hàm kích hoạt mà trả về giá trị output .

Về cơ bản, với một tập *N* bộ dữ liệu input , thuật toán thực hiện 4 bước:

1. Bước feedforward: Với một bộ dữ liệu input , tính giá trị đầu ra của network, trong quá trình tính toán, lưu lại các activation *a(l)* tại mỗi layer*.*
2. Tính đạo hàm riêng giá trị hàm lỗi cho mỗi nơ-ron thứ *j* ở output layer .
3. Cập nhật trọng số và bias ở output layer.
4. Làm tương tự bước 2 và bước 3 cho các layer *l = L-1, L-2,…1*, với L = Lhidden+1, Lhidden là số layer ẩn của mô hình.
5. Các tham số dùng để xây dựng mô hình được cài đặt theo hàm API của thư viện sklearn như bảng sau.

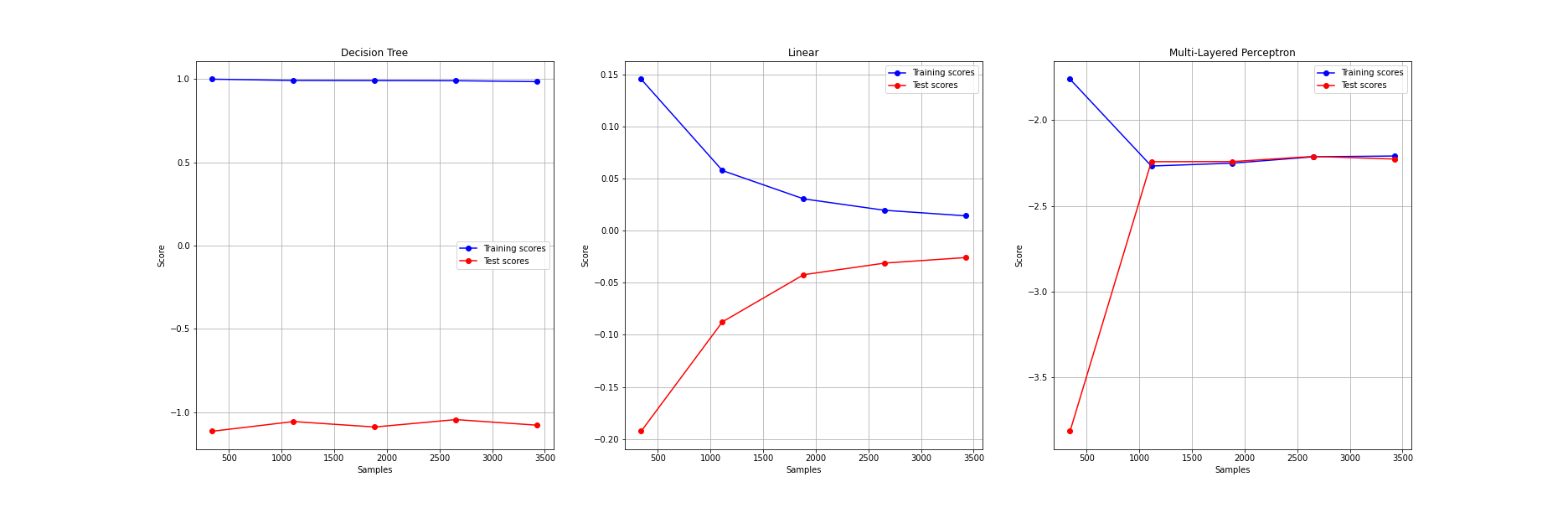
|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên tham số | Giá trị | Mô tả |
| 1 | hidden\_layer\_sizes | (64,64,64) | Chỉ định kích thước các lớp perceptron ẩn. |
| 2 | activation | 'relu' | Chỉ định hàm kích hoạt được sử dụng. |
| 3 | solver | 'adam' | Chỉ định hàm tối ưu trọng số. |
| 4 | alpha | 0.0001 | Tham số thuật toán L2 penalty. |
| 5 | batch\_size | 'auto' | Chỉ định số batch hoặc mặc định là min(200, số mẫu input). |
| 6 | learning\_rate | 'constant' | Chỉ định tính chất của tốc độ học. |
| 7 | learning\_rate\_init | 0.001 | Tốc độ học ban đầu. |
| 8 | max\_iter | 8572 | Số vòng lặp tối đa của thuật toán hồi quy. |
| 9 | shuffle | True | Chỉ định có tráo đổi thứ tự các mẫu dữ liệu train hay không. |
| 10 | random\_state | None | Điều khiển tính ngẫu nhiên khi khởi tạo trọng số và bias. |
| 11 | verbose | False | Chỉ định in tiến độ quá trình huấn luyện hay không. |
| 12 | warm\_start | False | Chỉ định sử dụng lại kết quả của lần huấn luyện trước hay không. |
| 13 | early\_stopping | False | Chỉ định sử dụng tham số ‘tol’ để kết thúc hồi quy sớm hay không. |
| 14 | beta\_1 | 0.9 | Tốc độ decay của vector mô-men thứ nhất khi dùng hàm tối ưu trọng số ‘adam’. |
| 15 | beta\_2 | 0.999 | Tốc độ decay của vector mô-men thứ hai khi dùng hàm tối ưu trọng số ‘adam’. |
| 16 | epsilon | 1e-08 | Giá trị ổn định số học của thuật toán ‘adam’. |

Bảng . Tham số mô hình MLP.

## Biểu đồ hiệu suất train, validate của từng mô hình

Biểu đồ hiệu suất trên bộ dữ liệu train và validate, được tạo bằng cách chia bộ dữ liệu train ban đầu thành 5 phần, mỗi phần có kích thước lần lượt bằng 0.1, 0.33, 0.55, 0.78, 1.0 lần bộ dữ liệu train. Sau đó thuật toán 5-fold cross-validation [4] được áp dụng cho mỗi phần để vẽ biểu đồ hiệu suất ở hình 1. Chi tiết kích thước bộ dữ liệu train và validate của quá trình được tính theo bảng 3.

Hình . Biểu đồ train, validate của các mô hình.

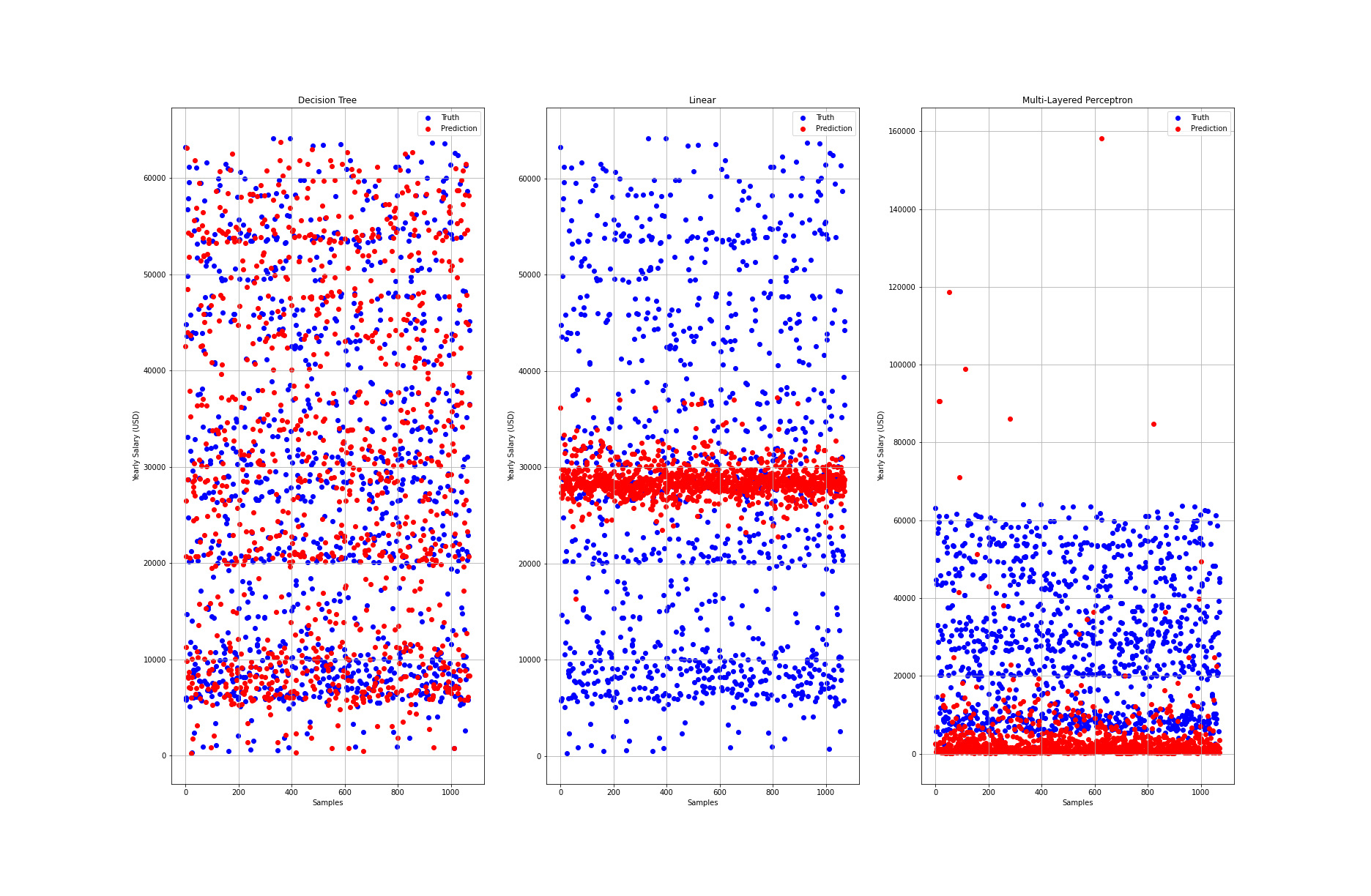


|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Kích thước bộ dữ liệu train = 4286* | | | | |
| STT phần | Tỷ lệ kích thước | Kích thước bộ dữ liệu | Kích thước bộ dữ liệu train | Kích thước bộ dữ liệu validate |
| 1 | 0.1 | 428 | 342 | 86 |
| 2 | 0.33 | 1414 | 1114 | 283 |
| 3 | 0.55 | 2357 | 1885 | 472 |
| 4 | 0.78 | 3343 | 2656 | 669 |
| 5 | 1.0 | 4286 | 3428 | 858 |

Bảng . Chi tiết kích thước bộ dữ liệu train, validate.

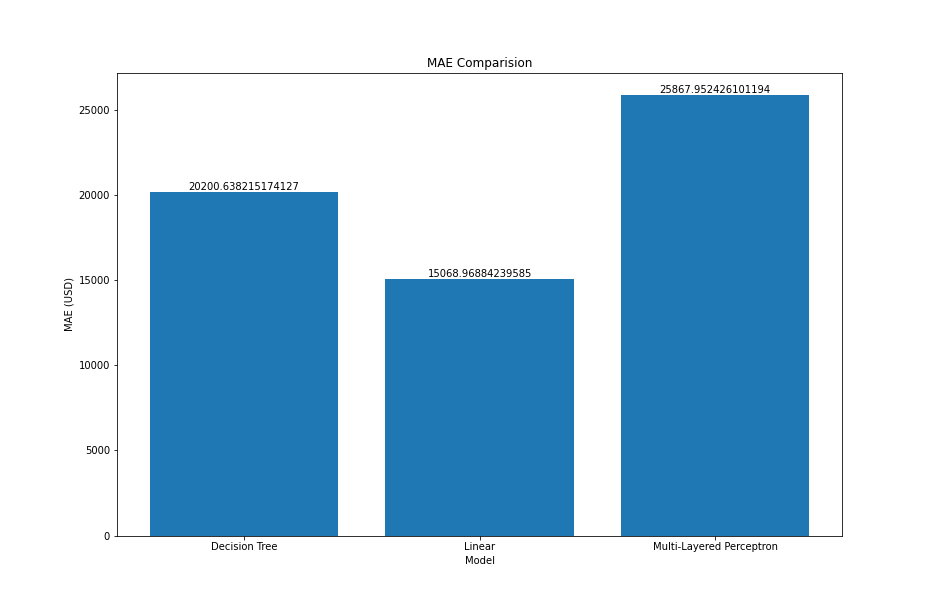
## Kết quả huấn luyện

Kết quả dự đoán từ 1072 mẫu được lấy từ bộ dữ liệu test được vẽ trên biểu đồ scatter ở hình 3.



Hình . Biểu đồ so sánh kết quả dự đoán

Sai số tuyệt đối trung bình (mean absolute error, MAE) của mỗi mô hình dựa trên 1072 mẫu trên được vẽ trên biểu đồ cột ở hình 3.



Hình . Biểu đồ sai số giữa của cácmô hình.

# Kết luận

* Mô hình hồi quy tuyến tính hoạt động tốt nhất với sai số tuyệt đối trung bình là 15,068.969 (USD); mô hình mạng nơ-ron nhân tạo nhiều lớp hoạt động kém nhất với sai số tuyệt đối trung bình là 25,867.952 (USD). Mô hình cây quyết định dự đoán tốt trên bộ dữ liệu train nhưng rất tệ trên bộ dữ liệu test.

# Tài liệu tham khảo

1. Chauhan, N. S. (2020). *Decision Tree Algorithm, Explained*. Kdnuggets. <https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html>.
2. Wikipedia contributors. (2021, July 18). Ordinary least squares. In *Wikipedia, The Free Encyclopedia*. Retrieved 14:24, August 4, 2021, from <https://en.wikipedia.org/w/index.php?title=Ordinary_least_squares&oldid=1034182640>.
3. Tiep, V. H. (2017, Feb 24). *Bài 14: Multi-layer Perceptron và Backpropagation*. Machine Learning cơ bản. <https://machinelearningcoban.com/2017/02/24/mlp/>.
4. Brownlee, J. (2018, May 23). *A Gentle Introduction to k-fold Cross-Validation*. Machine Learning Mastery. <https://machinelearningmastery.com/k-fold-cross-validation/>.