# 3.6. Dự đoán giá nhà: Một ví dụ về hồi quy

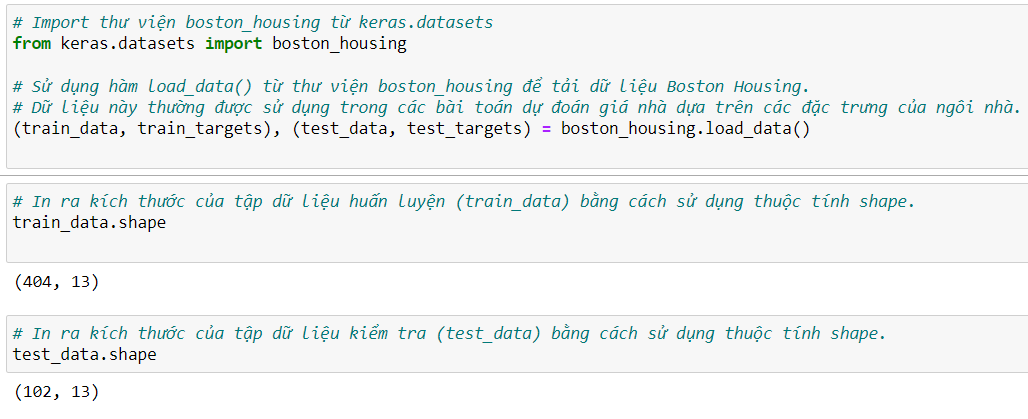
Hai ví dụ trước đã được coi là các vấn đề về phân loại, trong đó mục tiêu là dự đoán một nhãn rời rạc duy nhất của một điểm dữ liệu đầu vào. Một loại thông thường khác của vấn đề học máy là hồi quy, trong đó bao gồm việc dự đoán một giá trị liên tục thay vì một nhãn rời rạc: ví dụ, dự đoán nhiệt độ ngày mai, dựa trên dữ liệu khí tượng; hoặc dự đoán thời gian mà một dự án phần mềm sẽ mất để hoàn thành, dựa trên thông số kỹ thuật của nó.

* ***LƯU Ý:*** Đừng nhầm lẫn giữa hồi quy và thuật toán hồi quy logistic *(logistic regression)*. Một cách gây nhầm lẫn, hồi quy logistic không phải là một thuật toán hồi quy - nó là một thuật toán phân loại.

## 3.6.1. Bộ dữ liệu Giá nhà tại Boston

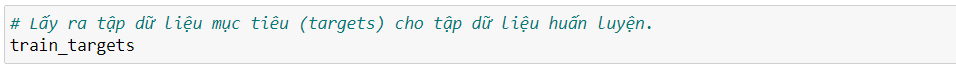
Bạn sẽ cố gắng dự đoán giá trung bình của các căn nhà trong một khu vực ngoại ô tại Boston vào giữa những năm 1970, dựa trên các điểm dữ liệu về khu vực đó vào thời điểm đó, như tỷ lệ tội phạm, thuế bất động sản địa phương và nhiều thông tin khác. Bộ dữ liệu bạn sử dụng có một điểm khác biệt thú vị so với hai ví dụ trước. Nó chỉ có một số ít điểm dữ liệu: chỉ có 506 điểm, được chia thành 404 mẫu huấn luyện và 102 mẫu kiểm tra. Và mỗi đặc trưng trong dữ liệu đầu vào (ví dụ: tỷ lệ tội phạm) có một tỷ lệ khác nhau. Ví dụ, một số giá trị là tỷ lệ, chúng có giá trị từ 0 đến 1; những giá trị khác nằm trong khoảng từ 1 đến 12, còn lại nằm giữa 0 và 100, và còn nhiều giá trị khác nữa.

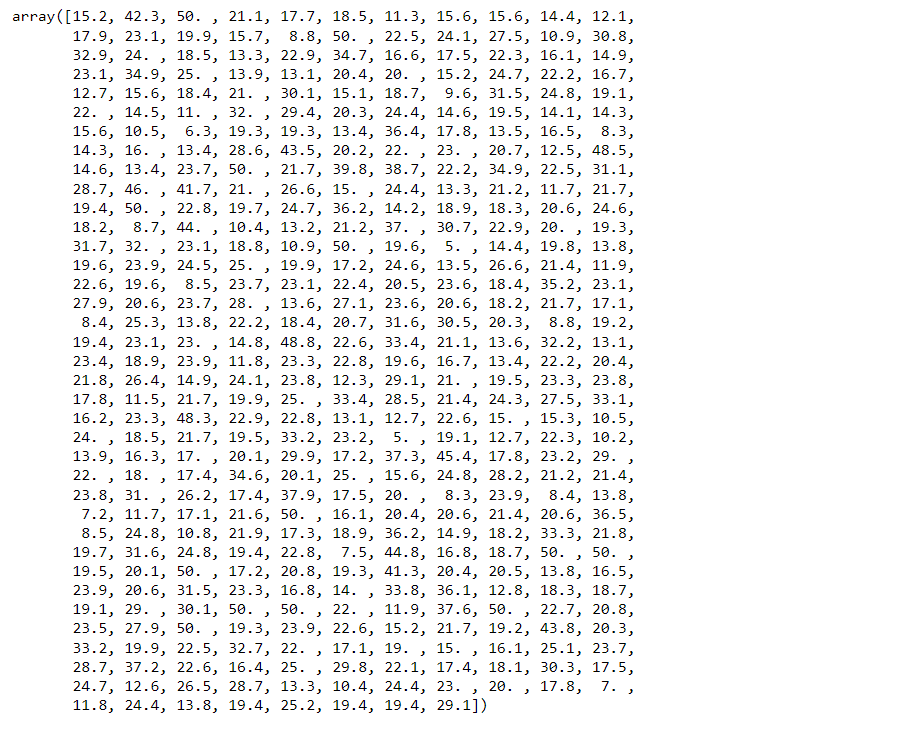
* ***Listing 3.24 Loading the Boston housing dataset***



Như bạn có thể thấy, bạn có 404 mẫu huấn luyện và 102 mẫu kiểm tra, mỗi mẫu có 13 đặc trưng số học khác nhau, như tỷ lệ tội phạm trên mỗi người, số phòng trung bình trên mỗi căn nhà, khả năng tiếp cận các con đường cao tốc, và nhiều đặc trưng khác.

Mục tiêu của bạn là dự đoán giá trị trung bình của các căn nhà do chủ sở hữu đang ở, tính bằng nghìn đô la:

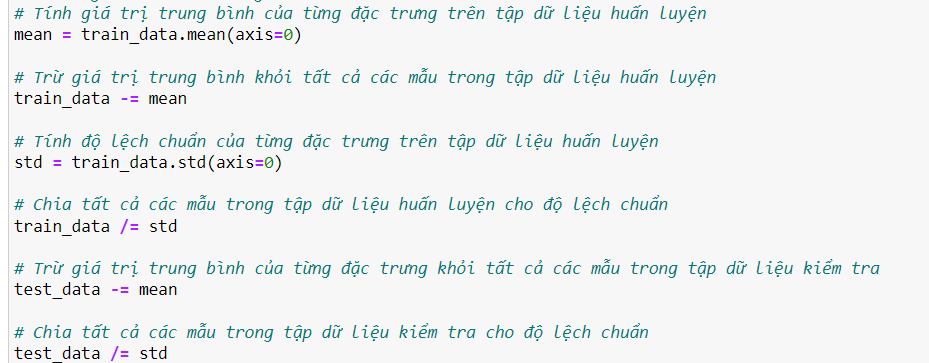


Giá trị trung bình của các căn nhà thường nằm trong khoảng từ 10,000 đô la đến 50,000 đô la. Nếu bạn thấy giá này thấp, hãy nhớ rằng đây là giữa những năm 1970, và giá này chưa được điều chỉnh cho lạm phát.

## 3.6.2. Chuẩn bị dữ liệu

Sẽ rất khó khăn nếu đưa vào mạng neural các giá trị có phạm vi rất khác nhau như vậy. Mạng có thể tự động thích nghi với dữ liệu đa dạng như vậy, nhưng điều này nhất định sẽ làm cho quá trình học tập khó khăn hơn. Một thực hành phổ biến để xử lý dữ liệu như vậy là chuẩn hóa theo từng đặc trưng: đối với mỗi đặc trưng trong dữ liệu đầu vào (một cột trong ma trận dữ liệu đầu vào), bạn trừ đi giá trị trung bình của đặc trưng và chia cho độ lệch chuẩn, để đặc trưng này có giá trị trung tâm xung quanh 0 và có độ lệch chuẩn đơn vị. Điều này có thể thực hiện dễ dàng trong Numpy.

* ***Listing 3.25 Normalizing the data***

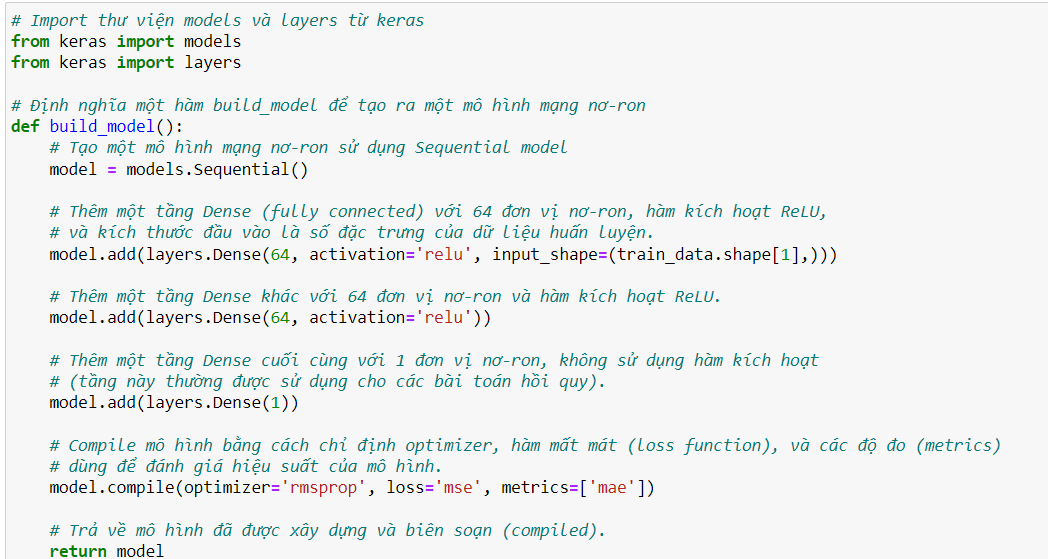


Lưu ý rằng các giá trị được sử dụng để chuẩn hóa dữ liệu kiểm tra được tính toán bằng cách sử dụng dữ liệu huấn luyện. Bạn không nên sử dụng bất kỳ giá trị nào được tính toán trên dữ liệu kiểm tra trong quy trình làm việc của bạn, ngay cả đối với việc chuẩn hóa dữ liệu đơn giản như vậy.

## 3.6.3. Xây dựng mạng của bạn

Bởi vì có rất ít mẫu dữ liệu sẵn có, bạn sẽ sử dụng một mạng rất nhỏ với hai lớp ẩn, mỗi lớp có 64 đơn vị. Nhìn chung, càng ít dữ liệu huấn luyện bạn có, càng nghiêm trọng hiện tượng quá khớp sẽ xảy ra, và việc sử dụng một mạng nhỏ là một cách để giảm thiểu hiện tượng quá khớp.

* ***Listing 3.26 Model definition***



Mạng kết thúc với một đơn vị duy nhất và không có hàm kích hoạt (nó sẽ là một lớp tuyến tính). Điều này là cài đặt điển hình cho hồi quy số học (một hồi quy nơi bạn cố gắng dự đoán một giá trị liên tục duy nhất). Áp dụng một hàm kích hoạt sẽ giới hạn phạm vi mà đầu ra có thể nhận được; ví dụ, nếu bạn áp dụng một hàm kích hoạt sigmoid cho lớp cuối cùng, mạng chỉ có thể học cách dự đoán các giá trị nằm giữa 0 và 1. Ở đây, vì lớp cuối cùng là tuyến tính hoàn toàn, mạng có thể tự do học cách dự đoán các giá trị trong bất kỳ phạm vi nào.

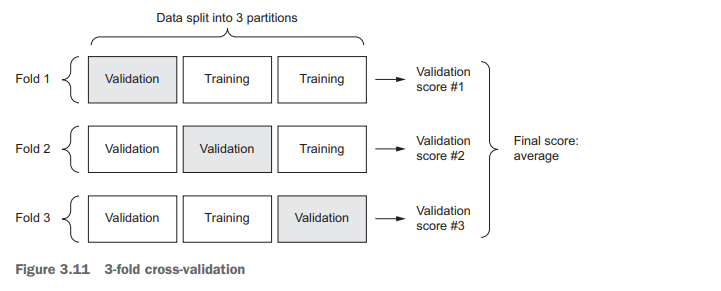
Lưu ý rằng bạn biên dịch mạng với hàm mất mát MSE - mean squared error, tức là bình phương của sự khác biệt giữa các dự đoán và mục tiêu. Đây là một hàm mất mát được sử dụng rộng rãi cho các vấn đề hồi quy.

Bạn cũng đang theo dõi một chỉ số mới trong quá trình huấn luyện: mean absolute error (MAE) - đây là giá trị tuyệt đối của sự khác biệt giữa các dự đoán và mục tiêu. Ví dụ, một MAE là 0.5 trong vấn đề này sẽ có nghĩa là trung bình các dự đoán của bạn sai lệch 500 đô la.

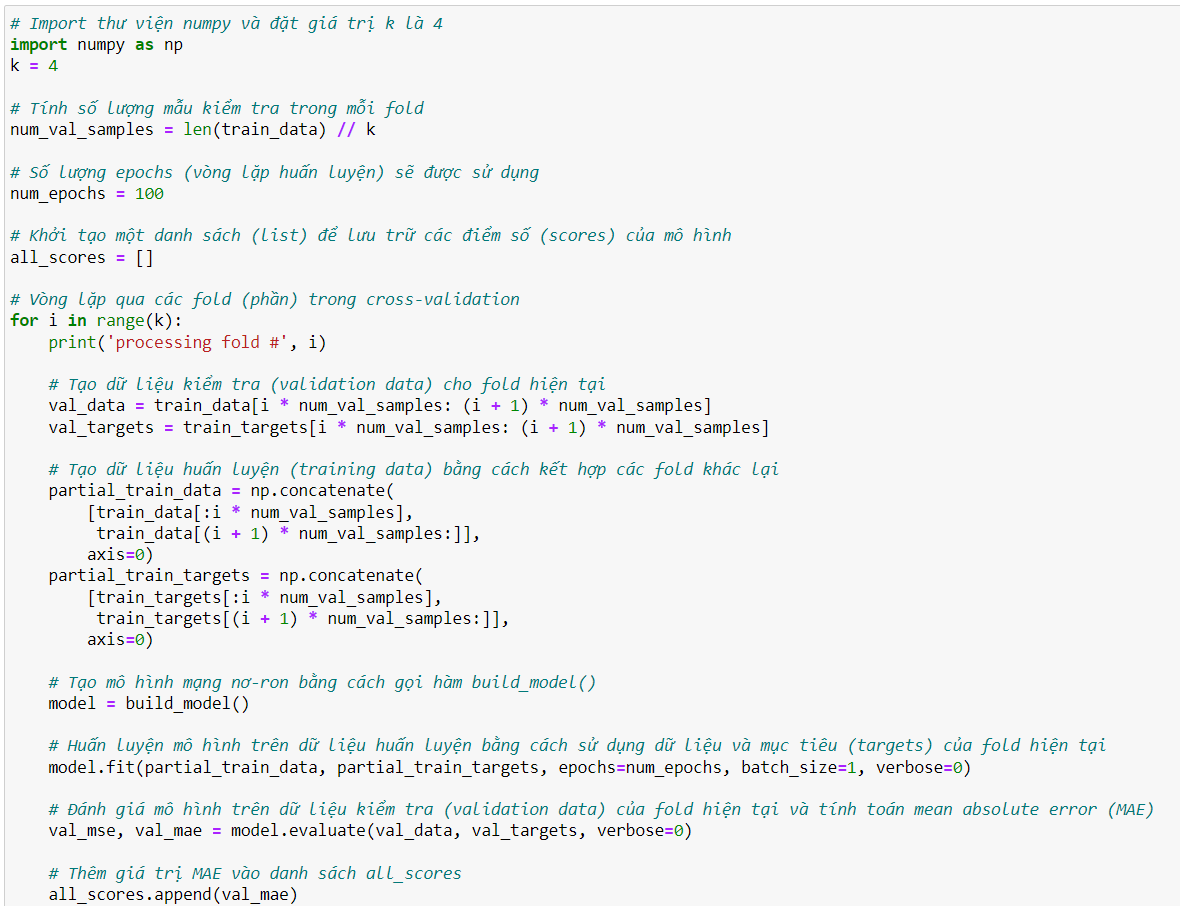
## 3.6.4. Xác thực phương pháp của bạn bằng cách sử dụng kiểm tra K-fold.

Để đánh giá mạng của bạn trong khi bạn tiếp tục điều chỉnh các thông số của nó (như số lần lặp để huấn luyện), bạn có thể chia dữ liệu thành một tập huấn luyện và một tập kiểm tra, như bạn đã làm trong các ví dụ trước. Nhưng vì bạn chỉ có rất ít điểm dữ liệu, tập kiểm tra cuối cùng sẽ rất nhỏ (ví dụ, khoảng 100 ví dụ). Do đó, điểm số kiểm tra có thể thay đổi nhiều tùy thuộc vào bạn chọn sử dụng những điểm dữ liệu nào cho việc kiểm tra và những điểm nào cho việc huấn luyện: điểm số kiểm tra có thể có phương sai cao đối với phân chia kiểm tra. Điều này sẽ ngăn bạn đánh giá mô hình một cách đáng tin cậy.

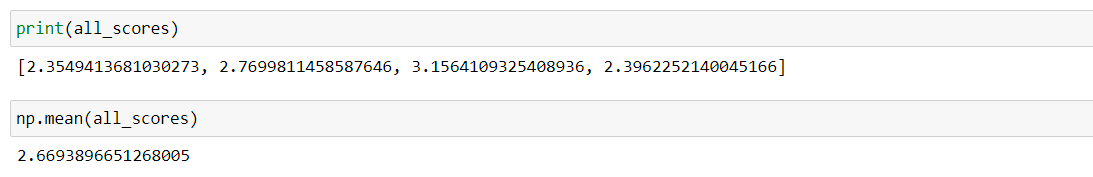
Thực hành tốt nhất trong tình huống như vậy là sử dụng phương pháp đánh giá K-fold cross-validation (xem hình 3.11). Nó bao gồm việc chia dữ liệu có sẵn thành K phần (thường K = 4 hoặc 5), tạo K mô hình giống hệt nhau và huấn luyện mỗi mô hình trên K - 1 phần trong khi đánh giá trên phần còn lại. Điểm số kiểm tra cho mô hình được sử dụng sau đó là trung bình của K điểm số kiểm tra thu được. Về mặt mã lệnh, điều này khá đơn giản.



* ***Listing 3.27 K-fold validation***

******

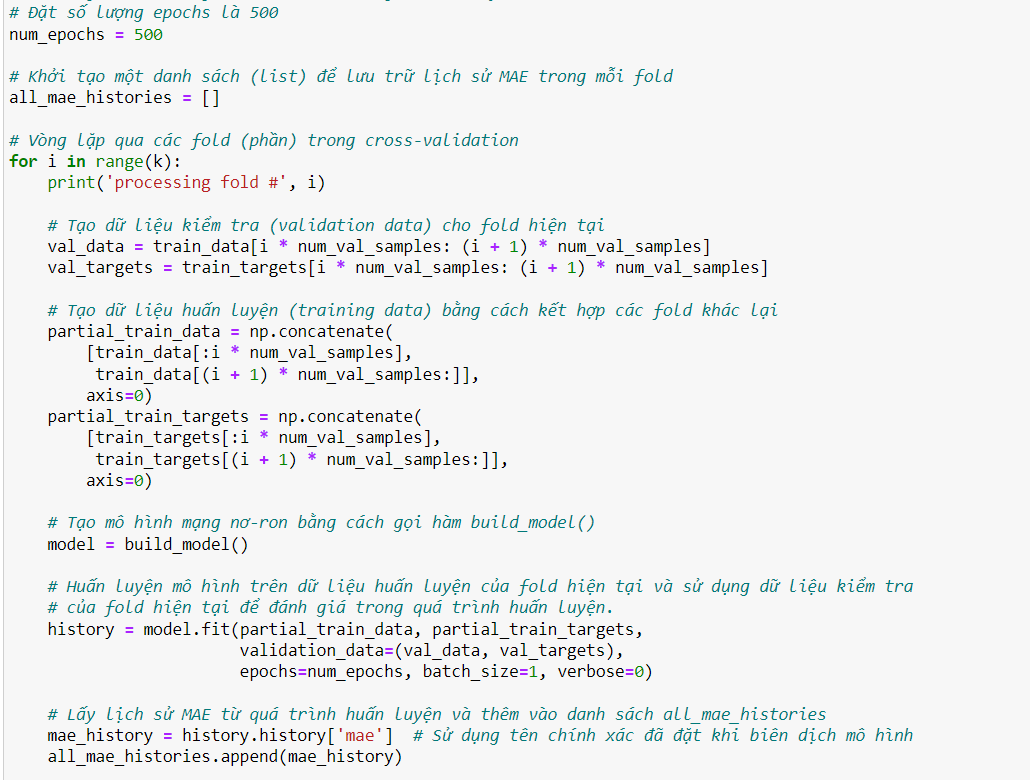
******



Các chạy khác nhau thực sự cho thấy các điểm số kiểm tra khác nhau, từ 2.6 đến 3.2. Trung bình (3.0) là một chỉ số đáng tin cậy hơn nhiều so với bất kỳ điểm số duy nhất nào - đó là mục đích chính của K-fold cross-validation. Trong trường hợp này, trung bình sai lệch là 3,000 đô la trên một căn nhà trung bình, điều này có ý nghĩa đáng kể khi xem xét khoảng giá từ 10,000 đến 50,000 đô la.

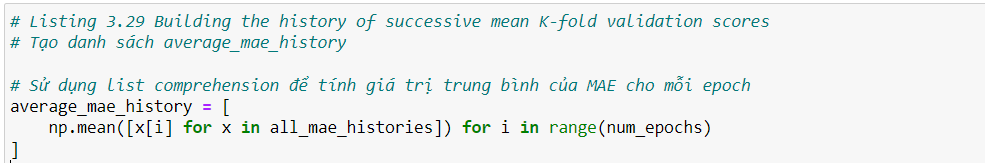
Hãy thử huấn luyện mạng thêm một chút: 500 epochs. Để theo dõi kết quả của mô hình sau mỗi epoch, bạn sẽ điều chỉnh vòng lặp huấn luyện để lưu lại nhật ký điểm số kiểm tra sau mỗi epoch.

* ***Listing 3.28 Saving the validation logs at each fold***



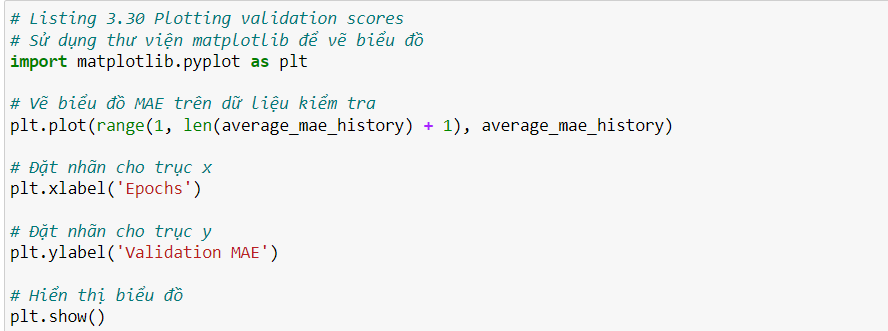
Sau đó, bạn có thể tính trung bình của các điểm số MAE từng epoch cho tất cả các phân vùng.

* ***Listing 3.29 Building the history of successive mean K-fold validation scores***

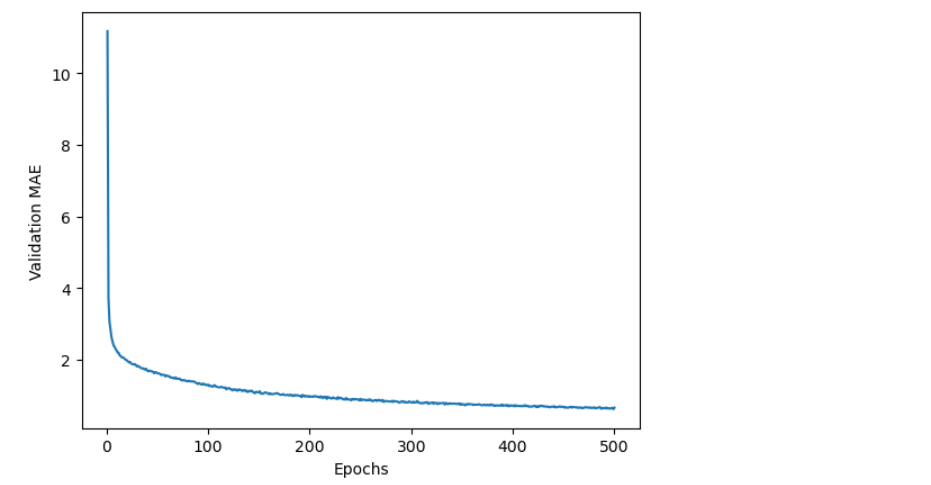


Chúng ta hãy vẽ đồ thị này; xem hình 3.12.

* ***Listing 3.30 Plotting validation scores***



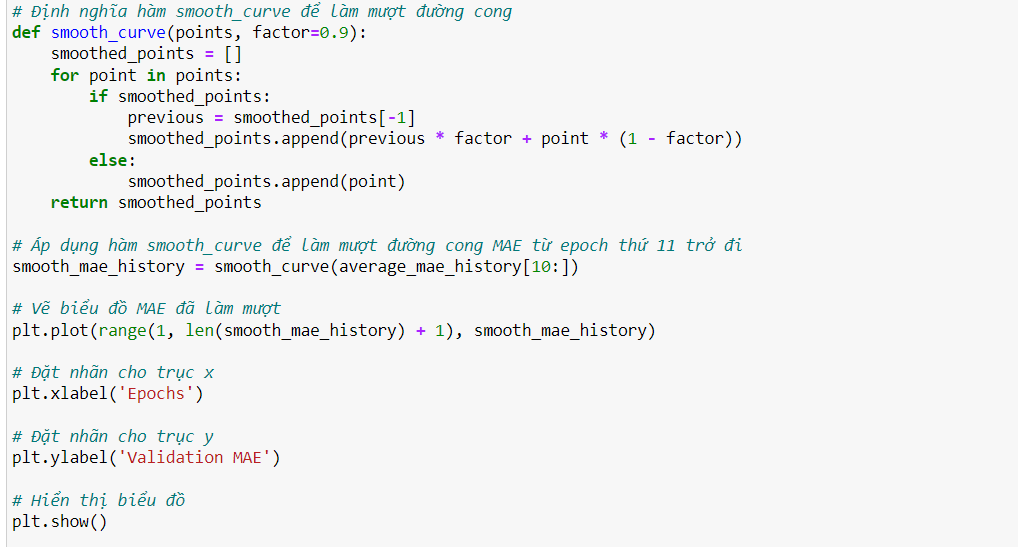
* ***Figure 3.12 Validation MAE by epoch***

  
Có thể khó nhìn thấy biểu đồ này một chút do vấn đề về tỷ lệ và phương sai tương đối cao. Chúng ta hãy thực hiện các bước sau để cải thiện:

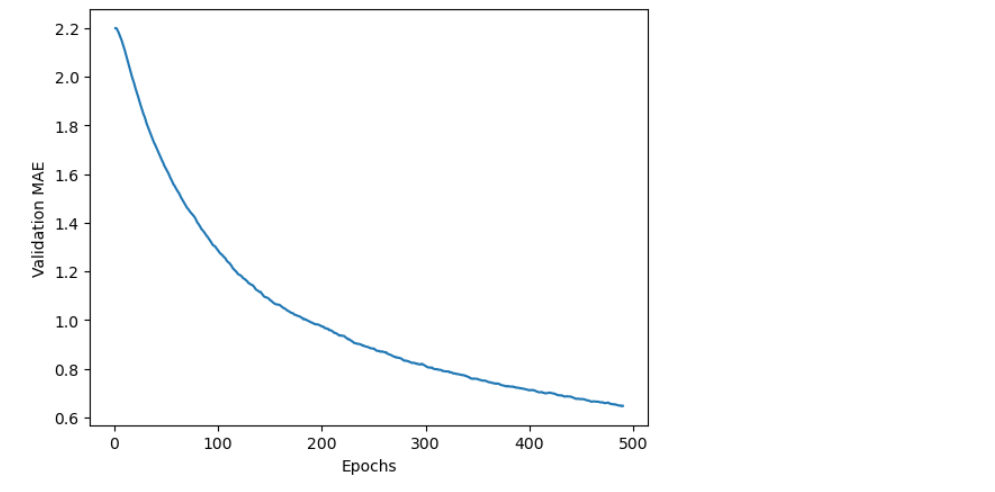
* Loại bỏ 10 điểm dữ liệu đầu tiên, vì chúng có tỷ lệ khác với phần còn lại của đường cong.
* Thay thế mỗi điểm bằng một trung bình di chuyển mũi tên của các điểm trước đó, để có được một đường cong mượt mà hơn.

Kết quả được hiển thị trong hình 3.13.

* ***Listing 3.31 Plotting validation scores, excluding the first 10 data points***



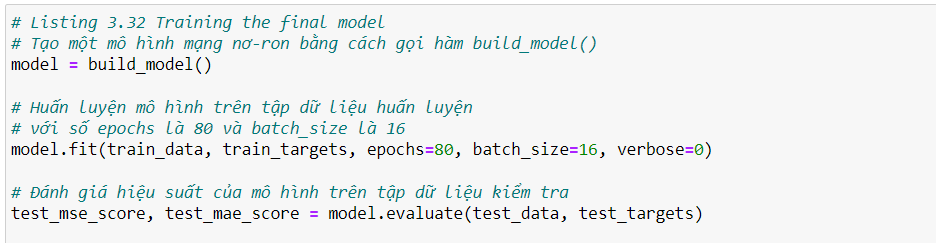
* ***Figure 3.13 Validation MAE by epoch, excluding the first 10 data points***

******

Dựa trên biểu đồ này, sai số trung bình của kiểm tra (validation MAE) ngừng cải thiện đáng kể sau 80 epoch. Sau điểm đó, bạn bắt đầu gặp hiện tượng quá khớp.

Khi bạn đã điều chỉnh các thông số khác của mô hình (ngoài số epoch, bạn cũng có thể điều chỉnh kích thước của các lớp ẩn), bạn có thể huấn luyện một mô hình sản xuất cuối cùng trên toàn bộ dữ liệu huấn luyện, với các tham số tốt nhất, và sau đó xem xét hiệu suất của nó trên dữ liệu kiểm tra.

* ***Listing 3.32 Training the final model***

******

Dưới đây là kết quả cuối cùng:

Bạn vẫn sai lệch khoảng $2,550.

## 3.6.5. Kết thúc

Dưới đây là những điểm quan trọng bạn nên rút ra từ ví dụ này:

* Hồi quy được thực hiện bằng cách sử dụng các hàm mất mát khác so với những gì chúng ta sử dụng cho phân loại. Mean squared error (MSE) là một hàm mất mát thường được sử dụng cho hồi quy.
* Tương tự, các phương pháp đánh giá được sử dụng cho hồi quy khác với các phương pháp được sử dụng cho phân loại; tự nhiên, khái niệm về độ chính xác không áp dụng cho hồi quy. Một phương pháp đánh giá phổ biến cho hồi quy là mean absolute error (MAE).
* Khi các đặc trưng trong dữ liệu đầu vào có giá trị trong các phạm vi khác nhau, mỗi đặc trưng nên được chuẩn hóa độc lập như một bước tiền xử lý.
* Khi có ít dữ liệu có sẵn, sử dụng K-fold validation là một cách tốt để đánh giá một cách đáng tin cậy một mô hình.
* Khi có ít dữ liệu huấn luyện có sẵn, nên ưu tiên sử dụng một mạng nhỏ với ít lớp ẩn (thường chỉ một hoặc hai lớp), để tránh hiện tượng quá khớp nghiêm trọng.

# 3.7. Tóm tắt chương

* Bây giờ bạn đã có khả năng xử lý các loại nhiệm vụ học máy phổ biến trên dữ liệu vector: phân loại nhị phân, phân loại nhiều lớp và hồi quy số học. Các phần "Kết thúc" ở phần đầu chương tóm tắt các điểm quan trọng mà bạn đã học về những loại nhiệm vụ này.
* Thông thường, bạn sẽ cần tiền xử lý dữ liệu thô trước khi đưa vào mạng neural.
* Khi dữ liệu của bạn có các đặc trưng với phạm vi khác nhau, hãy tỉ lệ lại từng đặc trưng một độc lập như một phần của quá trình tiền xử lý.
* Khi quá trình huấn luyện diễn ra, các mạng neural cuối cùng sẽ bắt đầu quá khớp và đạt được kết quả tồi hơn trên dữ liệu chưa từng thấy trước đó.
* Nếu bạn không có nhiều dữ liệu huấn luyện, hãy sử dụng một mạng nhỏ với chỉ một hoặc hai lớp ẩn để tránh quá khớp nghiêm trọng.
* Nếu dữ liệu của bạn được chia thành nhiều danh mục, bạn có thể gây ra hiện tượng hạn chế thông tin nếu bạn làm cho các lớp trung gian quá nhỏ.
* Hồi quy sử dụng các hàm mất mát và các phương pháp đánh giá khác so với phân loại.
* Khi bạn làm việc với ít dữ liệu, kiểm tra K-fold có thể giúp đánh giá mô hình của bạn một cách đáng tin cậy.