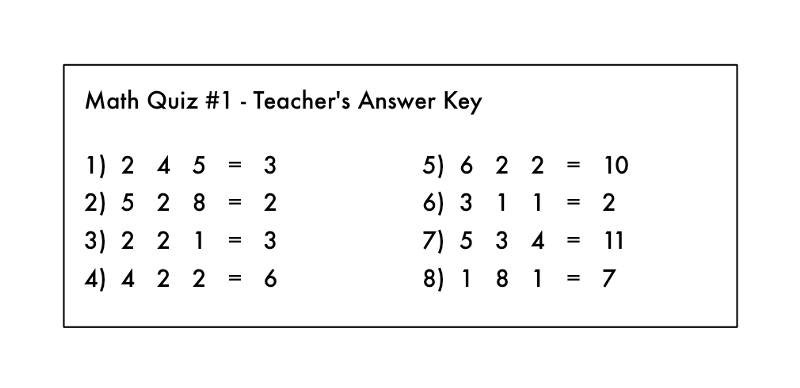
1. **Vấn đề đặt ra**

* Giả sử bạn là một người làm về bất động sản. Công ty của bạn phát triển nhanh, bạn tuyển hàng loạt thực tập. Nhưng có một vấn đề - bạn có thể dễ dàng nhìn lướt qua và đánh giá chính xác giá trị của một ngôi nhà, nhưng với các thực tập viên không có kinh nghiệm, họ không biết mỗi căn trị giá bao nhiêu.
* Để giúp đỡ các bạn thực tập này, bạn quyết định viết một ứng dụng nhỏ có thể ước lượng được giá của một ngôi nhà dựa vào diện tích, khu vực lân cận, ...
* Và bạn ghi lại thông tin của mọi căn nhà được bán trong thành phố, trong vòng 3 tháng. Với mỗi ngôi nhà, bạn ghi lại mọi thông tin: số phòng ngủ, diện tích (feet vuông), neighborhood, ... Nhưng quan trọng nhất là giá (price) cuối cùng của căn nhà được bán:



* Chúng ta có được "training data" , Với dữ liệu "training data" như trên, chúng ta muốn viết một ứng dụng có thể ước tính được giá của một căn nhà tương tự khác:



* Chúng ta muốn sử dụng training data để dự đoán giá của những ngôi nhà khác.
* Đây được gọi là supervised learning. Bạn biết được giá mỗi căn nhà được bán đi, nói cách khác, bạn biết được câu trả lời của bài toán, và có thể từ đây suy ra được logic của vấn đề.
* Để xây dựng ứng dụng này, bạn cho training data về mỗi ngôi nhà này vào một thuật toán machine learning. Thuật toán sẽ cố gắng tìm ra loại tính toán nào để các con số có thể work.
* Nó giống như việc tìm các toán tử trong bài tập toán hồi lớp 1 chúng ta vẫn hay được học:
* Từ bảng trên, bạn có thể tìm ra được các phép toán nào để có được kết quả bên phải? Bạn biết bạn có nghĩa vụ phải "làm gì đó" với những con số ở bên trái để có được câu trả lời ở bên phải.

1. **Thuật toán áp dụng vào đề tài : Supervised Learning (Học có giám sát)**

* Supervised learning là thuật toán dự đoán đầu ra (outcome) của một dữ liệu mới (new input) dựa trên các cặp (input, outcome) đã biết từ trước. Cặp dữ liệu này còn được gọi là (data, label), tức (dữ liệu, nhãn). Supervised learning là nhóm phổ biến nhất trong các thuật toán Machine Learning.
* cách toán học, Supervised learning là khi chúng ra có một tập hợp biến đầu vào X={x1,x2,…,xN}X={x1,x2,…,xN} và một tập hợp nhãn tương ứng Y={y1,y2,…,yN}Y={y1,y2,…,yN}, trong đó xi,yixi,yi là các vector. Các cặp dữ liệu biết trước (xi,yi)∈X×Y(xi,yi)∈X×Y được gọi là tập training data (dữ liệu huấn luyện). Từ tập traing data này, chúng ta cần tạo ra một hàm số ánh xạ mỗi phần tử từ tập XX sang một phần tử (xấp xỉ) tương ứng của tập YY:
* yi≈f(xi),  ∀i=1,2,…,Nyi≈f(xi),  ∀i=1,2,…,NMục đích là xấp xỉ hàm số ff thật tốt để khi có một dữ liệu xx mới, chúng ta có thể tính được nhãn tương ứng của nó y=f(x)y=f(x).
* Thuật toán gồm 2 loại : Classification (Phân loại) , Regression (Hồi quy)
* **Regression (Hồi quy)**
* Linear Regression
* Logistic Regression
* Stepwise Regression
* **Classification (Phân loại)**
* Linear Classifier
* Support Vector Machine (SVM)
* Kernel SVM
* Sparse Representation-based classification (SRC)

1. Mô tả chi tiết thuật toán
2. **Dữ liệu đầu vào:**

* Đã có một tập dữ liệu gồm các thuộc tính của bất động sản và giá của chúng
* Các thuộc tính như: số phòng ngủ, số phòng tắm, vị trí địa lý, năm xây dựng...và giá bán
* Giờ muốn tìm một công thức nào đó để từ một bất động sản bất kì, cho biết một vài thông tin như số phòng ngủ, vị trí địa lý... có thể tính toán ra được giá bán

1. **Hướng giải quyết:**

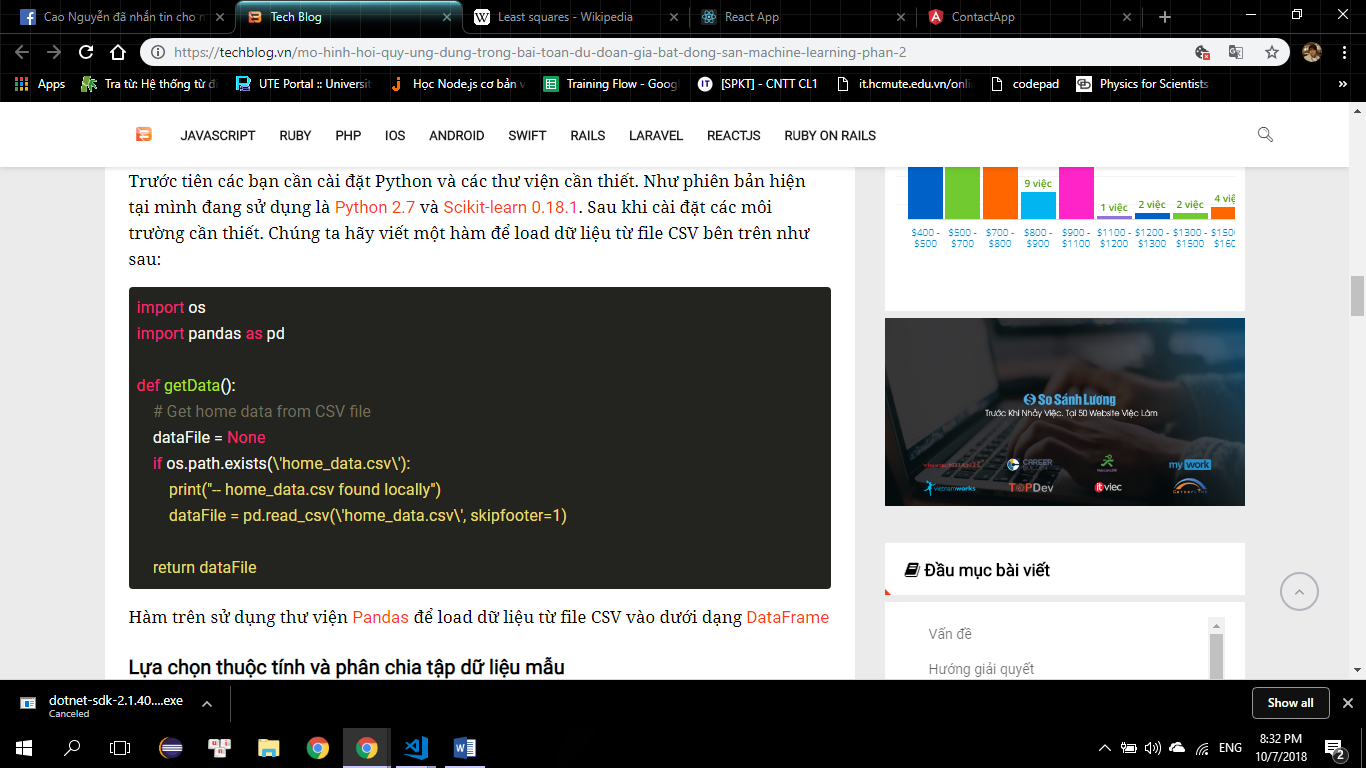
* Bản chất của hồi quy chính là tìm ra mối quan hệ nào đó giữa **biến phụ thuộc** (dependence) Y (ví dụ như kết quả xổ số hay giá nhà) và một hay nhiều **biến độc lập** X
* Đối với mô hình hồi quy tuyến tính, mô hình hóa sử dụng là dạng bậc nhất. Có nghĩa là chúng ta cần tính toán các hệ số Wi trong một biểu thức bậc nhất như sau: *Y = W0 + W1.X1 + W2.X2 + ... + Wn.Xn*
* Tức là trên đồ thị chúng ta đi tìm một đường thẳng (nếu trường hợp có 1 biến độc lập) hoặc một siêu phẳng (với trường hợp nhiều biến độc lập) đi qua tập hợp các điểm trong không gian thuộc tính mà thể hiện **gần đúng nhất** sự phân bố của tập dữ liệu. Trên phương diện tính toán, chúng ta đi tìm các hệ số Wi như trên sao cho **lỗi hồi quy**đạt được là nhỏ nhất. Lại nói về lỗi hồi quy người ta thường sử dụng **Phương pháp bình phương nhỏ nhất** (least square).

1. **Ứng dụng vào bài toán dự đoán giá bất động sản**

* Đầu tiên chúng ta cần phải có một tập dữ liệu File CSV này chứa thông tin về các bất động sản như số phòng ngủ, số phòng tắm, năm xây dựng... và giá bán tương ứng của nó. Chúng ta sẽ áp dụng lý thuyết về mô hình hồi quy để từ tập dữ liệu này, xây dựng một hàm sử dụng để định giá cho một bất động sản bất kì trong tương lai

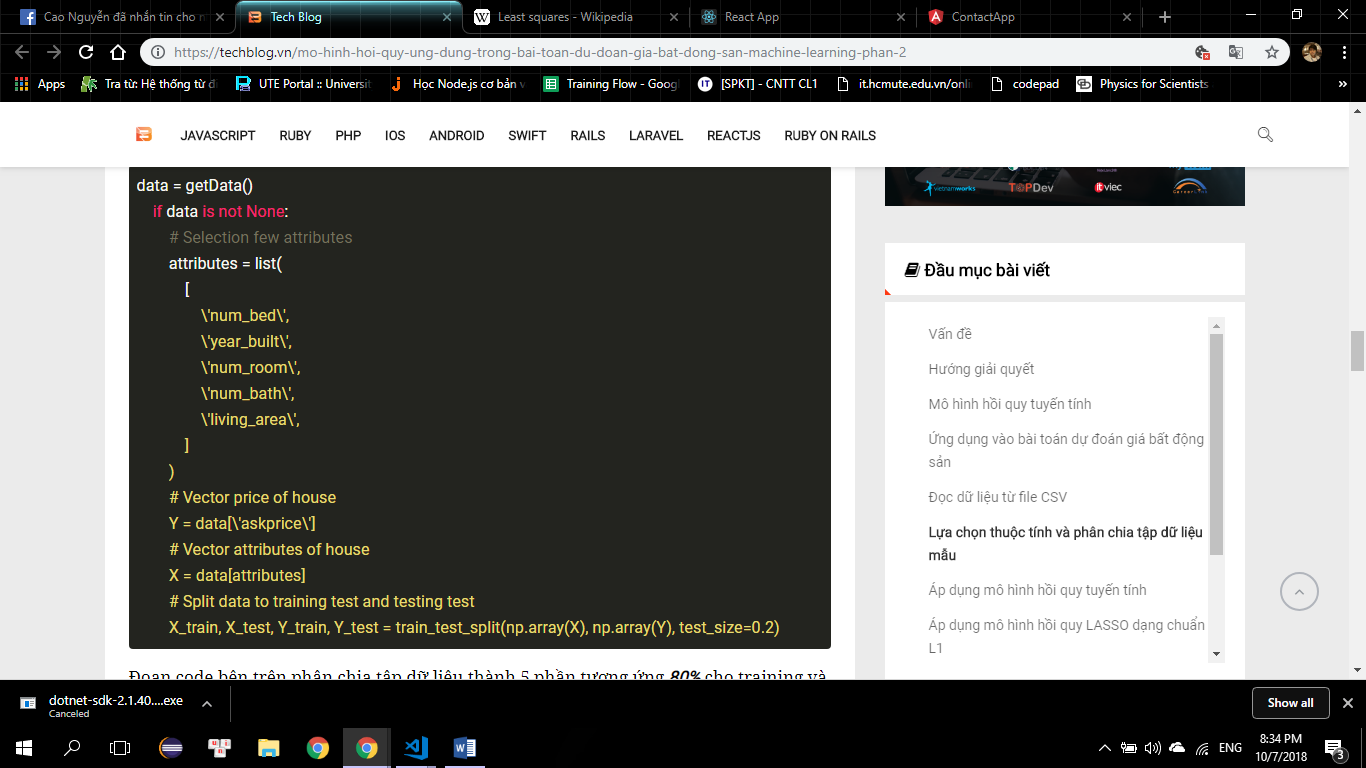
### **Đọc dữ liệu từ file CSV**

Trước tiên các bạn cần cài đặt Python và các thư viện cần thiết. Như phiên bản hiện tại mình đang sử dụng là [Python 2.7](https://www.python.org/download/releases/2.7/) và [Scikit-learn 0.18.1](http://scikit-learn.org/stable/support.html). Sau khi cài đặt các môi trường cần thiết. Chúng ta hãy viết một hàm để load dữ liệu từ file CSV bên trên như sau:



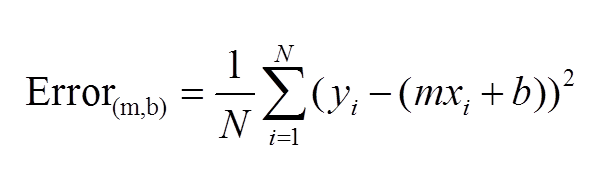
Hàm trên sử dụng thư viện [Pandas](http://pandas.pydata.org/) để load dữ liệu từ file CSV vào dưới dạng [DataFrame](http://pandas.pydata.org/pandas-docs/stable/dsintro.html)

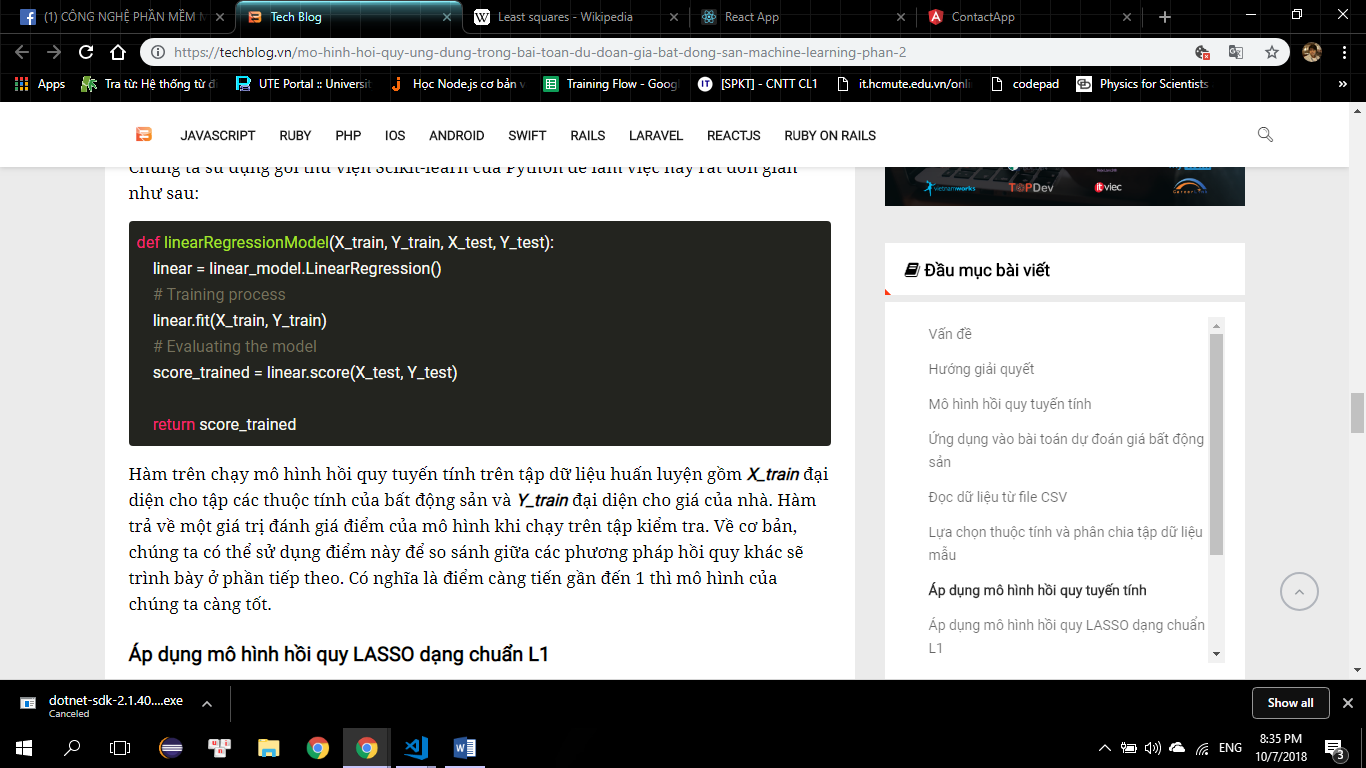
### Lựa chọn thuộc tính và phân chia tập dữ liệu mẫu

Tư tưởng của chúng ta là sẽ phân chia tập dữ liệu mẫu thành hai tập con là tập dữ liệu huấn luyện và tập dữ liệu kiểm tra. Việc này sử dụng tư tưởng của kiểm tra chéo (cross validation). Ngoài ra, trong tập dữ liệu mẫu có rất nhiều thuộc tính có ý nghĩa và có thể khai thác thêm, ví dụ như từ kinh độ và vĩ độ chúng ta có thể tìm thêm các thuộc tính như khoảng cách trung tâm thành phố, số bệnh viện lân cận...

Đoạn code bên trên phân chia tập dữ liệu thành 5 phần tương ứng **80%** cho training và **20%** cho testing. Việc cần làm tiếp theo là viết một hàm chạy Phương pháp hồi quy tuyến tính.

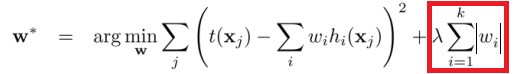
### Áp dụng mô hình hồi quy tuyến tính

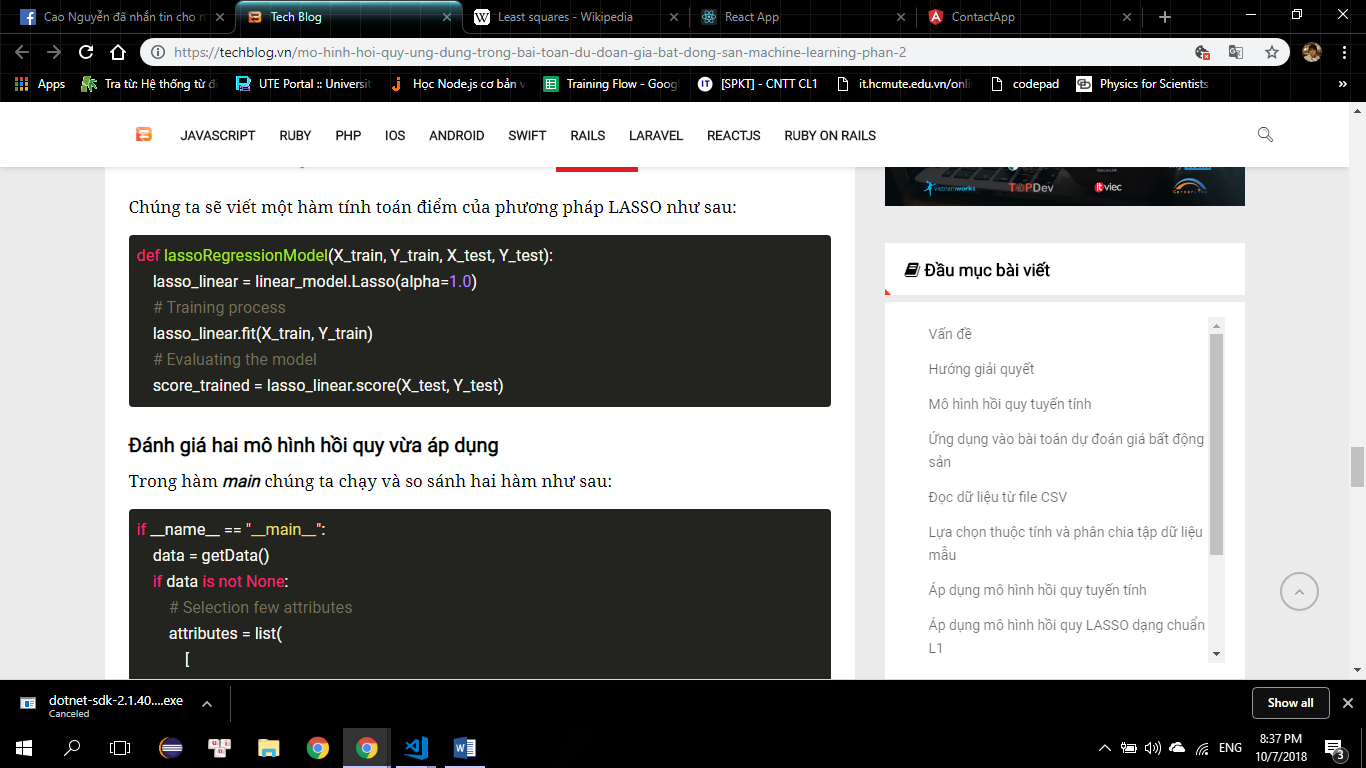
Về cơ bản, việc huấn luyện theo mô hình tuyến tính bản chất là đi tìm các giá trị **m** và **b** sao cho cực tiểu hóa hàm lỗi sau:  


Chúng ta sử dụng gói thư viện Scikit-learn của Python để làm việc này rất đơn giản như sau:

Hàm trên chạy mô hình hồi quy tuyến tính trên tập dữ liệu huấn luyện gồm **X\_train** đại diện cho tập các thuộc tính của bất động sản và **Y\_train** đại diện cho giá của nhà. Hàm trả về một giá trị đánh giá điểm của mô hình khi chạy trên tập kiểm tra. Về cơ bản, chúng ta có thể sử dụng điểm này để so sánh giữa các phương pháp hồi quy khác sẽ trình bày ở phần tiếp theo. Có nghĩa là điểm càng tiến gần đến 1 thì mô hình của chúng ta càng tốt.

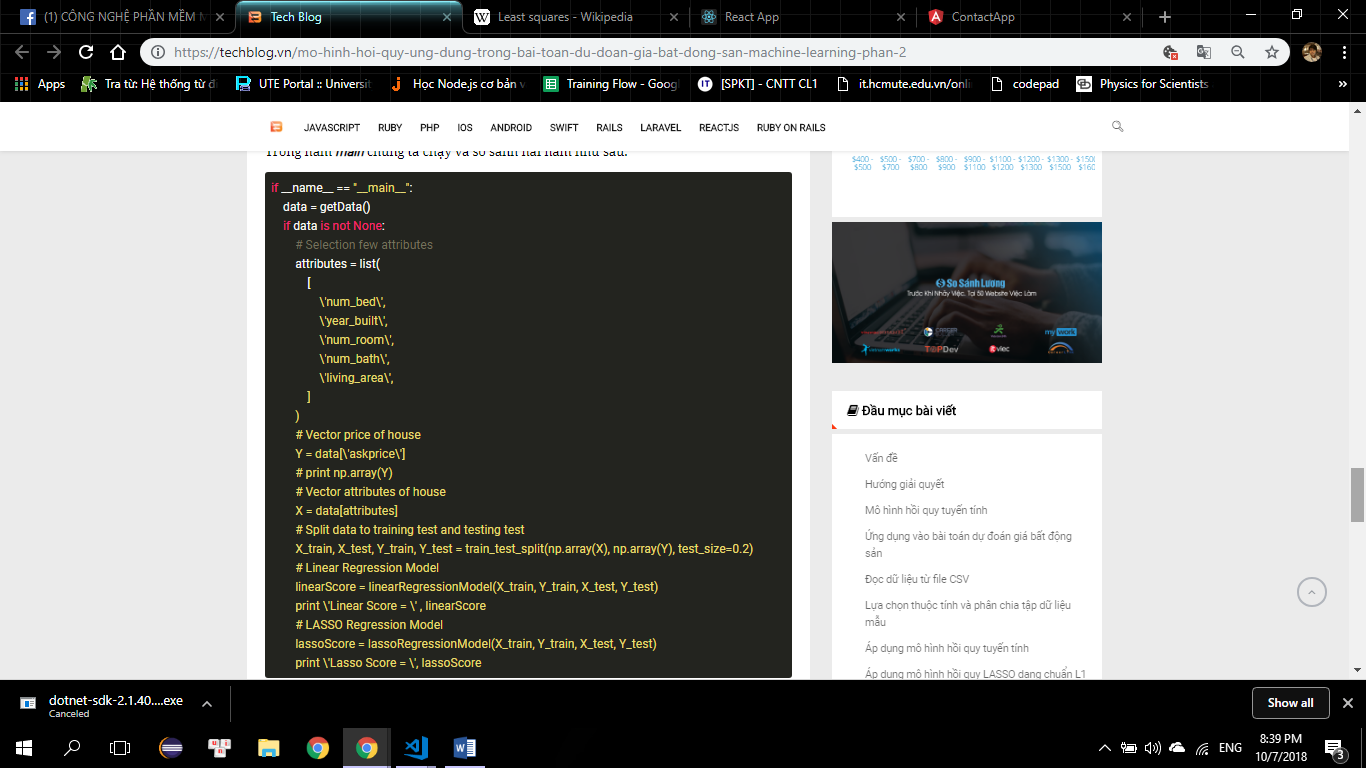
### Áp dụng mô hình hồi quy LASSO dạng chuẩn L1

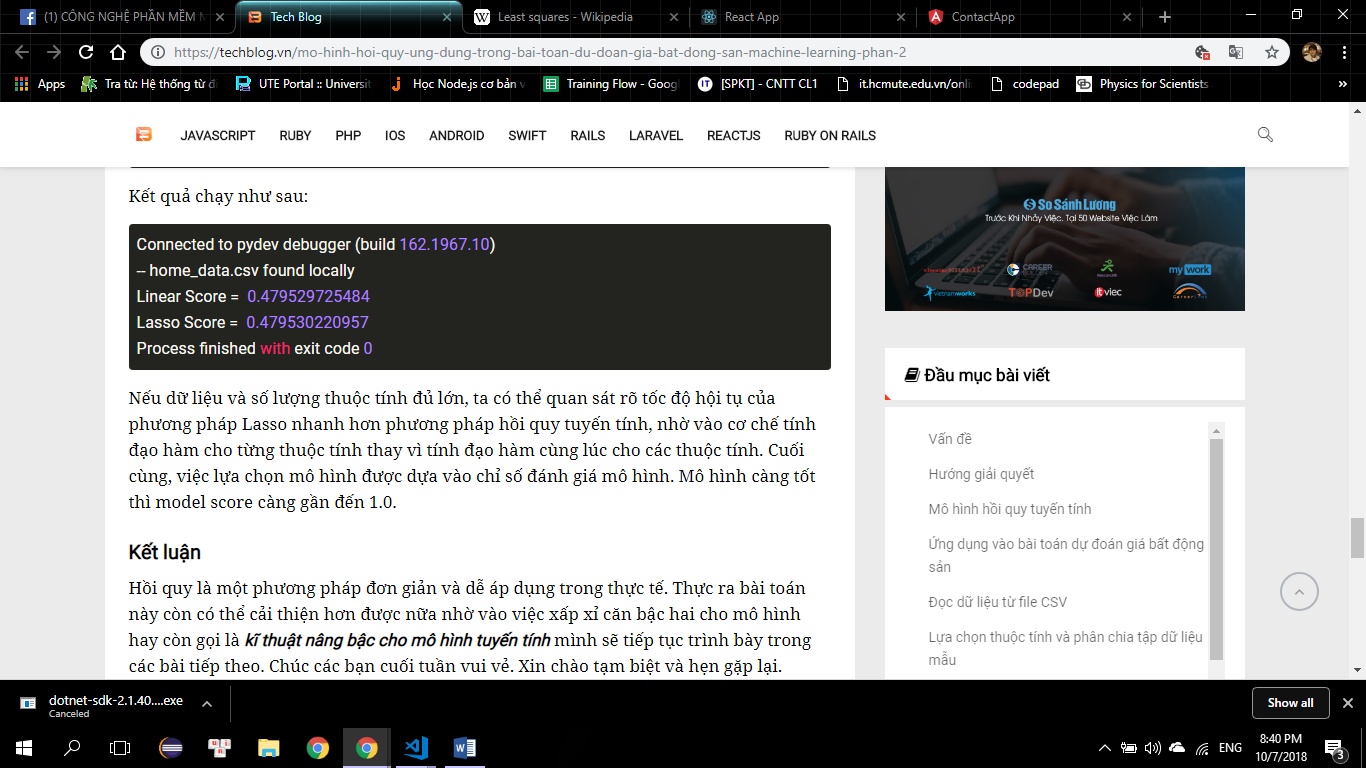
Các dạng chuẩn (Regularization) thường được đưa vào các mô hình để nhằm giảm thiểu hiện tượng **over-fitting**. tuyến tính một đại lượng phạt lỗi **lamda**. Từ đó mô hình của chúng ta sẽ tìm các tham số phù hợp sao cho cực tiểu hóa hàm lỗi như sau:

Chúng ta sẽ viết một hàm tính toán điểm của phương pháp LASSO như sau:

### Đánh giá hai mô hình hồi quy vừa áp dụng

Trong hàm **main** chúng ta chạy và so sánh hai hàm như sau:



Kết quả chạy như sau:

Nếu dữ liệu và số lượng thuộc tính đủ lớn, ta có thể quan sát rõ tốc độ hội tụ của phương pháp Lasso nhanh hơn phương pháp hồi quy tuyến tính, nhờ vào cơ chế tính đạo hàm cho từng thuộc tính thay vì tính đạo hàm cùng lúc cho các thuộc tính. Cuối cùng, việc lựa chọn mô hình được dựa vào chỉ số đánh giá mô hình. Mô hình càng tốt thì model score càng gần đến 1.0.

* Vì thế, chúng ta cần điều chỉnh weights, giống như việc "đi xuống thung lũng" trong đồ thị để tìm được điểm thấp nhất. Nếu như chúng ta điều chỉnh từng chút một và luôn đi xuống, ta sẽ tìm được điểm cực tiểu mà không cần phải thử quá nhiều weights.