Hàm [np.random.seed](https://docs.scipy.org/doc/numpy-1.13.0/reference/generated/numpy.random.seed.html)

Các ngôn ngữ lập trình nói chung không tạo ra các giá trị ‘thực sự ngẫu nhiên’. Thật vậy, nếu bạn mở python và bắt đầu với:

**>>>** **import** numpy **as** np

**>>>** np.random.rand()

thì kết quả luôn là các số giống nhau ở mỗi lần thử (bạn hãy thoát python và thử lại nhiều lần xem). Như trên máy tính của tôi, kết quả lúc nào cũng là 0.38919680466308004. *Như vậy, hàm ngẫu nhiên không thực sự sinh ra các giá trị ngẫu nhiên.* Tuy nhiên, nếu thực hiện hàm này rất nhiều lần, chúng ta sẽ thu được các các số nằm trong khoảng [0, 1) mà xác suất để một điểm nằm trong đoạn [a, b] với 0 <= a < b < 1 bằng b - a.

Hàm np.random.seed() là một hàm được coi như giúp khởi tạo các bộ sinh số ngẫu nhiên (random generator). Biến số trong seed thường là một số nguyên không âm 32 bit. Với các giá trị của biến số khác nhau thì sẽ cho ra các số ngẫu nhiên khác nhau.

Hàm số này được dùng để đối chiều kết quả trong các lần chạy khác nhau trong các bài toán Machine Learning. Rất nhiều các thuật toán Machine Learning chạy dựa trên việc tính toán ngẫu nhiên (ví dụ, [Stochastic Gradient Descent](https://machinelearningcoban.com/2017/01/16/gradientdescent2/#-stochastic-gradient-descent) được sử dụng rất nhiều trong các thuật toán tối ưu Neural Networks). Để đối chiếu kết quả trong nhiều lần chạy trên, người ta thường khởi tạo các random generator với các seed như nhau.

Các bạn có thể để ý thấy rằng trong các bài trước tôi thường dùng np.random.seed(). Việc đó để đảm bảo rằng kết quả bạn tìm được giống với kết quả trong code mẫu.

Trích từ https://machinelearningcoban.com/2017/10/20/fundaml\_vectors/