

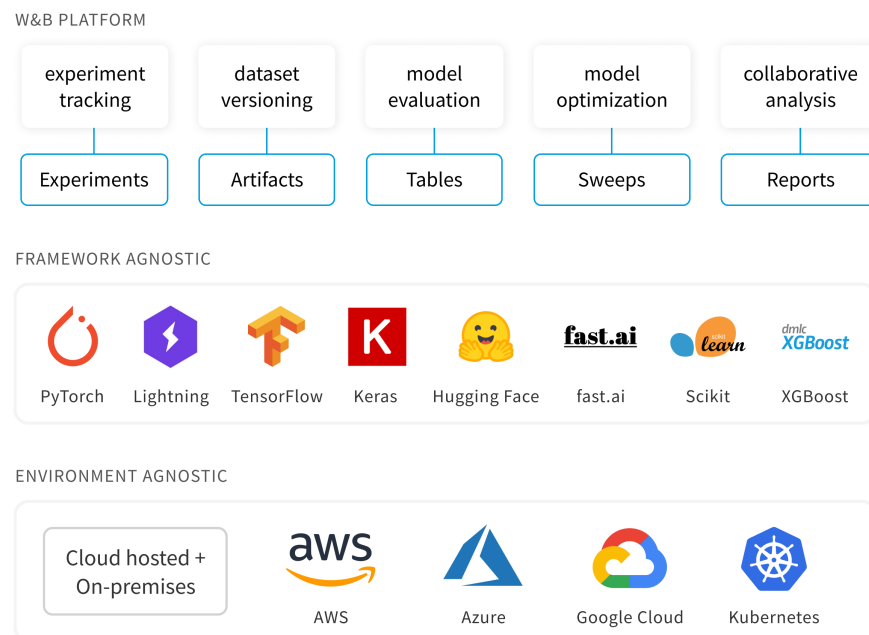
# Basic Python - Getting Started with WandB

*Hoàng-Nguyên Vũ và Quang-Vinh Đinh*



## 1. Mô tả:

- **WandB** hay còn gọi là **Weights and Biases** là một công cụ giúp các Data Scientist theo dõi mô hình, dữ liệu, thông tin hệ thống chỉ với vài dòng code. Weights & Bias hỗ trợ miễn phí cho các cá nhân hoặc tổ chức nghiên cứu. Công cụ này hỗ trợ rất nhiều nền tảng mà không cần chuyển đổi sang công cụ khác như TensorFlow, Keras, PyTorch, Sklearn, FastAI, ...
- Các thông tin cần quản lý và theo dõi sẽ được chuyển tới một giao diện (UI) của Weights & Biases. Giao diện này sẽ hiển thị và phân tích thông tin dễ dàng, việc so sánh các mô hình cũng như các tham số của nhiều thử nghiệm cũng diễn ra một cách nhanh chóng và trực tiếp. Một tiện ích nữa là bạn có thể chia sẻ các thông tin này cho team member của bạn. Dưới đây là một số công cụ trong WandB hỗ trợ bao gồm:

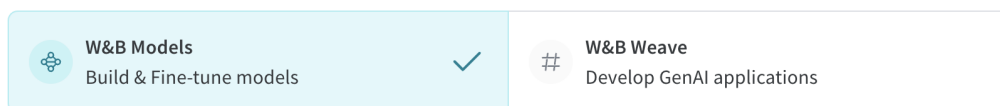


Hình 1: Tổng quan các công cụ của WandB.

- **Experiments:** Theo dõi các thử nghiệm trên các biểu đồ như evaluate function, loss function; thông tin về hệ thống như dung lượng bộ nhớ, thông tin xử lý GPU, ...
- **Artifacts:** Phiên bản dữ liệu, các phiên bản mô hình
- **Table:** Hiển thị các thông tin về các thử nghiệm và giá trị của các tham biến.
- **Sweeps:** Tối ưu hóa các tham số
- **Report:** Lưu trữ và chia sẻ các số liệu để có thể tái tạo mô hình

## 2. Cách triển khai WandB:

- **Tạo tài khoản:** Để bắt đầu, chúng ta cần tạo một tài khoản trên trang của Weights & Biases [tại đây](#). Sau khi đăng ký thành công, tại màn hình chính của trang web, chúng ta sẽ được cấp một API key để đăng nhập vào và sử dụng thư viện WandB.



### Quickstart: Tracking your first run in Weights & Biases

Weights & Biases' tools make it easy for you to quickly track experiments, visualize results, spot regressions, and more. Simply put, Weights & Biases enables you to build better models faster and easily share findings with colleagues.

Visualize your model training with Python / Pytorch or Open in Colab

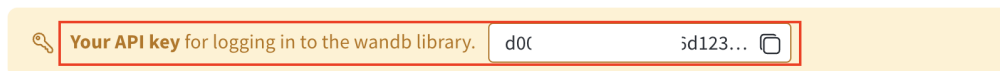
#### 1. Set up the wandb library

Install the CLI and Python library for interacting with the Weights and Biases API.

```
pip install wandb
```

Next, log in and paste your API key when prompted.

```
wandb login
```



Hình 2: Giao diện WandB sau khi đăng ký thành công.

- **Cài đặt và đăng nhập vào thư viện WandB:**
  - Để cài đặt thư viện WandB, chúng ta sẽ thực thi dòng lệnh sau và đăng nhập dựa trên API Key được cấp ở bước đăng ký ở trên.

```
1 # Cài đặt thư viện WandB
2 !pip install wandb
3
4 # Đăng nhập với API Key
5 !wandb login
```

```
1 !wandb login
```

wandb: Logging into wandb.ai. (Learn how to deploy a W&B server locally: <https://wandb.me/wandb-server>)  
wandb: You can find your API key in your browser here: <https://wandb.ai/authorize>  
wandb: Paste an API key from your profile and hit enter, or press ctrl+c to quit:

Hình 3: Đăng nhập vào WandB với API Key.

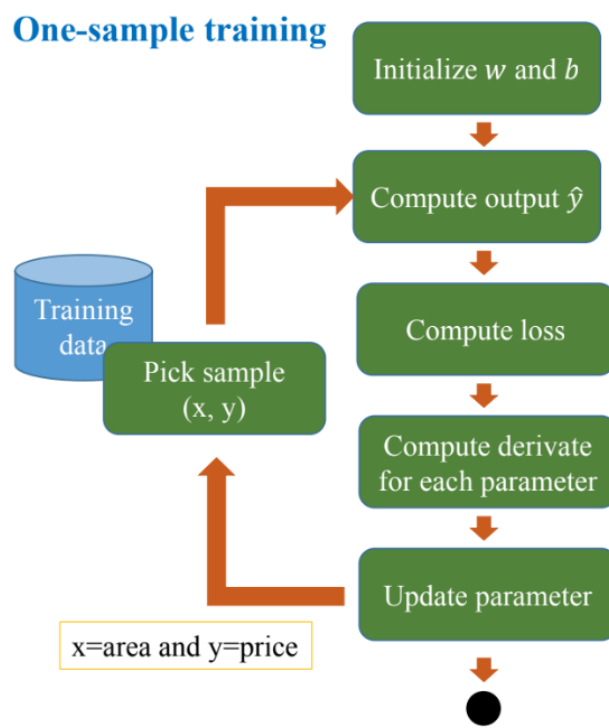
- **Làm quen với WandB:**

- Giả sử, chúng ta có tập dữ liệu giá nhà theo diện tích như bảng dưới, và chúng ta cần xây dựng mô hình hồi quy tuyến tính (Linear Regression) cơ bản để dự đoán giá nhà theo diện tích:

Bảng 1: Bảng dữ liệu giá nhà theo diện tích

Diện tích ( $m^2$ )	Giá
6.7	9.1
4.6	5.9
3.5	4.6
5.5	6.7

- Để xây dựng mô hình linear regression với tập dữ liệu trên, chúng ta sẽ xây dựng dựa theo lưu đồ dưới đây:



Hình 4: Lưu đồ cài đặt mô hình Linear Regression

Chúng ta sẽ cài đặt và theo dõi quá trình training của mô hình với WandB như sau:

### A. Import thư viện và khởi tạo dataset:

```
1 import pandas as pd
2 import wandb
3
4 areas = [6.7, 4.6, 3.5, 5.5]
5 prices = [9.1, 5.9, 4.6, 6.7]
6
7 dataset = pd.DataFrame({
8     'areas': areas,
9     'prices': prices
10 })
```

### B. Cài đặt mô hình Linear Regression kèm theo dõi quá trình huấn luyện với WandB:

#### + Cài đặt mô hình:

```
1 # forward
2 def predict(x, w, b):
3     return x*w + b
4
5 # compute gradient
6 def gradient(y_hat, y, x):
7     dw = 2*x*(y_hat-y)
8     db = 2*(y_hat-y)
9
10    return (dw, db)
11
12 # update weights
13 def update_weight(w, b, lr, dw, db):
14     w_new = w - lr*dw
15     b_new = b - lr*db
16
17    return (w_new, b_new)
```

#### + Huấn luyện mô hình và theo dõi quá trình với WandB:

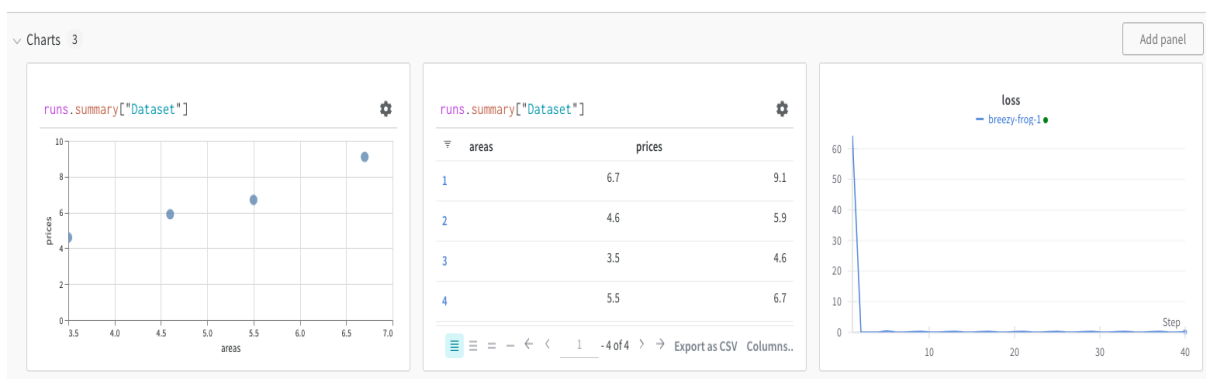
```
1 # init weights
2 b = 0.04
3 w = -0.34
4 lr = 0.01
5 epochs = 10
6
7 # init project wandb
8 wandb.init(
9     # Set the project where this run will be logged
10    project="demo-linear-regression",
11    config={
12        "learning_rate": lr,
13        "epochs": epochs,
14    },
15 )
```

```

16
17 wandb.run.log({"Dataset" : wandb.Table(dataframe=dataset)})
18
19 X_train = dataset['areas']
20 Y_train = dataset['prices']
21
22 N = len(X_train)
23 # parameter
24 losses = [] # for debug
25
26 for epoch in range(epochs):
27     # for an epoch
28     for i in range(N):
29         # get a sample
30         x = X_train[i]
31         y = Y_train[i]
32
33         # predict y_hat
34         y_hat = predict(x, w, b)
35
36         # compute loss
37         loss = (y_hat-y)*(y_hat-y) / 2.0
38
39         # tracking loss with WandB
40         wandb.log({"loss": loss})
41
42         # compute gradient
43         (dw, db) = gradient(y_hat, y, x)
44
45         # update weights
46         (w, b) = update_weight(w, b, lr, dw, db)
47
48 # Mark a run as finished, and finish uploading all data.
49 wandb.finish()

```

- Trong quá trình training mô hình, chúng ta có thể theo dõi trực tiếp tại trang web của WandB, với kết quả training khi thực thi dòng lệnh trên như sau:

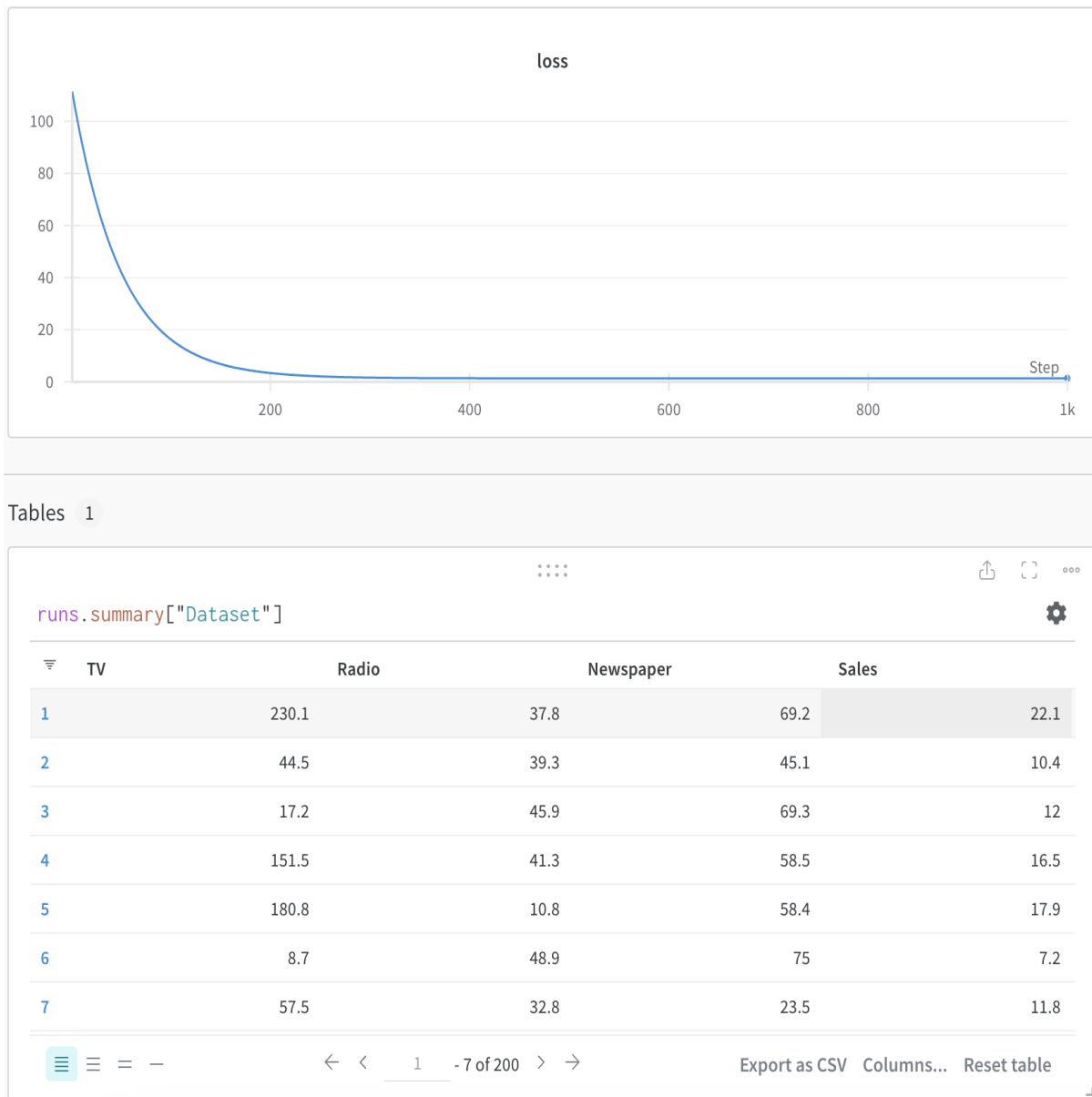


Hình 5: Kết quả theo dõi quá trình training mô hình Linear Regression

→ **Nhận xét:** Qua biểu đồ hàm loss được theo dõi ở trên, giúp chúng ta dễ dàng theo dõi quá trình training của mô hình.

3. **Bài tập:** Hãy đọc dữ liệu tại file sau: [advertising.csv](#). Xây dựng mô hình Linear Regression với 1000 epochs, khởi tạo các giá trị trọng số  $w_1 = w_2 = w_3 = 0$ ,  $b = 1$  và thực hiện tracking quá trình training với WandB cho tập data trên.

→ **Kết quả:**



Hình 6: Kết quả theo dõi quá trình training mô hình Linear Regression

- Hết -