Kaggle Is Everything You Need To build and train model

Trần Anh Tuấn

Ngày 17 tháng 5 năm 2024

Saigon University, Vietnam
Dasanbob22122002@gmail.com

Tóm tắt nội dung

Nhu cầu lập trình python bằng cách sử dụng các môi trường, cấu hình máy từ xa(remote) như Kaggle là 1 hướng tiếp cận mà gần như các nhà nghiên cứu ngày nay rất ưa thích sử dụng. Ngày nay việc chạy thực nghiệm mất hàng giờ, hàng ngày diễn ra thường xuyên, bởi thế có một yêu cầu rất nghiêm ngặt về việc chạy không gián đoạn. Bởi thế việc hiểu và sử dụng được các công cụ mà nền tảng Kaggle cho việc xây dựng và huấn luyện mô hình sẽ là một lợi thế lớn cho các công trình nghiên cứu khoa học.

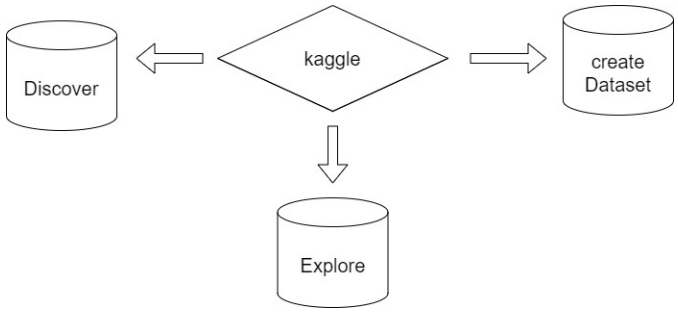
Từ khóa: Python, Kaggle, từ xa, chạy không gián đoạn, nghiên cứu khoa học.

1 Giới thiệu

Kaggle[1] là một nền tảng trực tuyến nổi tiếng trong cộng đồng khoa học dữ liệu và machine learning. Được thành lập vào năm 2010 và sau đó được Google mua lại vào năm 2017, Kaggle cung cấp một môi trường đa dạng cho các nhà khoa học dữ liệu, các nhà phân tích và các nhà phát triển machine learning.

Trên Kaggle, người dùng có thể tìm thấy hàng ngàn bộ dữ liệu từ nhiều lĩnh vực khác nhau, từ tài chính đến y tế và giáo dục. Nền tảng này cũng tổ chức các cuộc thi về khoa học dữ liệu và machine learning, nơi mà các thí sinh có thể tham gia để giải quyết các vấn đề thực tế và cạnh tranh để giành giải thưởng.

Ngoài ra, Kaggle cung cấp một công cụ Kernel, cho phép người dùng viết và chia sẻ mã Python (cũng như R) trong môi trường được quản lý. Điều này giúp tạo ra một cộng đồng mạnh mẽ của các nhà khoa học dữ liệu và machine learning, nơi mà họ có thể học hỏi từ nhau, chia sẻ kiến thức và cùng nhau giải quyết các vấn đề phức tạp. Kaggle đã trở thành một nguồn tài nguyên quý giá không chỉ cho những người mới bắt đầu mà còn cho các chuyên gia trong lĩnh vực này.



Hình 1: Minh họa về Kaggle

Bên cạnh Kaggle còn có các dịch vụ, nền tảng khác cũng cho sử dụng remote GPU như Google colab. Một trong những ưu điểm chính của Kaggle vượt trội so với Google

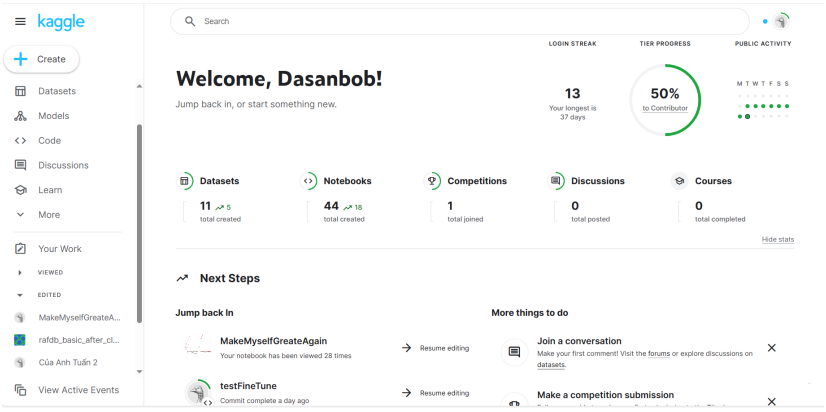
colab là sức mạnh của cộng đồng. Kaggle không chỉ là một nền tảng để chạy mã, mà còn là một cộng đồng lớn với hàng ngàn nhà khoa học dữ liệu và chuyên gia học máy. Điều này mang lại lợi ích đáng kể khi cần tìm kiếm hỗ trợ, chia sẻ kiến thức và thảo luận về các vấn đề liên quan đến dự án. Trong khi đó, Colab cũng có một cộng đồng nhưng không đồng đều và phong phú như Kaggle.

Một ưu điểm khác của Kaggle là việc cung cấp GPU miễn phí trong 30h/ tuần. Trong khi Colab cũng cung cấp dịch vụ GPU, nhưng có giới hạn về thời gian sử dụng và khả năng sử dụng GPU mạnh mẽ hơn. Trên Kaggle, bạn có thể sử dụng GPU miễn phí một cách linh hoạt và hiệu quả, giúp tăng tốc độ huấn luyện mô hình và xử lý dữ liệu lớn hơn.

2 Kiến Trúc của Kaggle

Kiến trúc của Kaggle bao gồm rất nhiều thành phần, nhưng nếu chỉ dừng lại ở mức xây dựng và huấn luyện mô hình thì nên tập trung vào notebooks và datasets

- **Platform Interface**: Giao diện người dùng trực quan và dễ sử dụng là một phần quan trọng của Kaggle. Nó cung cấp các công cụ để tìm hiểu, tham gia vào các cuộc thi, thảo luận với cộng đồng, và chia sẻ kiến thức.
- **Kernels(notebooks)**: Là một môi trường tính toán trực tuyến dựa trên web, Kernels cho phép người dùng viết và chạy mã Python, R, và các ngôn ngữ khác mà không cần cài đặt trên máy tính của họ. Điều này giúp các nhà khoa học dữ liệu và các nhà phân tích thống kê dễ dàng chia sẻ mã và phân tích của họ với cộng đồng. Hơn thế nữa chức năng Save Version trong notebooks giúp việc huấn luyện mô hình được diễn ra ngay cả khi thiết bị vật lý, mạng wifi bị tắt hoàn toàn.
- **Datasets**: Kaggle cung cấp một thư viện lớn các bộ dữ liệu miễn phí, từ dữ liệu thô đến dữ liệu được xử lý sẵn cho các dự án máy học và phân tích, thậm chí bất kì ai cũng có thể đăng tải dữ liệu nên ở mức cộng đồng hay cá nhân
- **Competitions (Cuộc thi)**: Kaggle tổ chức các cuộc thi máy học và khoa học dữ liệu định kỳ, thu hút các chuyên gia từ khắp nơi trên thế giới để tham gia giải quyết các vấn đề thực tế từ các doanh nghiệp hoặc tổ chức.
- **Notebooks và Discussion**: Kaggle cung cấp một nền tảng cho các nhà khoa học dữ liệu và nhà phân tích dữ liệu để chia sẻ ý tưởng, kỹ thuật và phân tích thông qua các tài liệu Notebook và diễn đàn thảo luận.
- **Community**: Kaggle có một cộng đồng lớn, đa dạng và chuyên nghiệp của các nhà khoa học dữ liệu và nhà phân tích dữ liệu, cung cấp cơ hội để học hỏi, hợp tác và chia sẻ kiến thức.

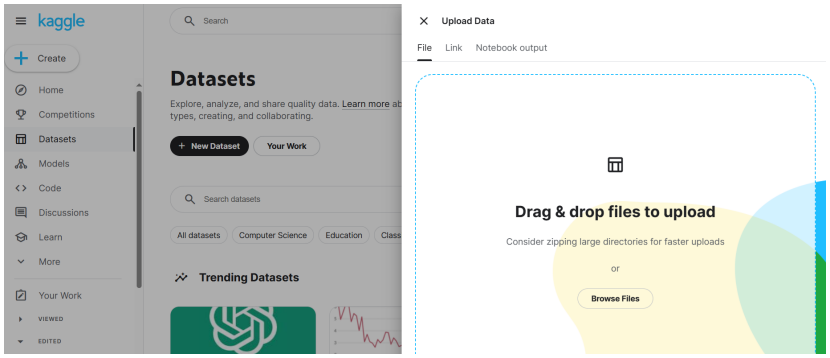


Hình 2: Kiến trúc mô hình của Git

3 Tất tần tật cách sử dụng Kaggle

3.1 Kaggle datasets

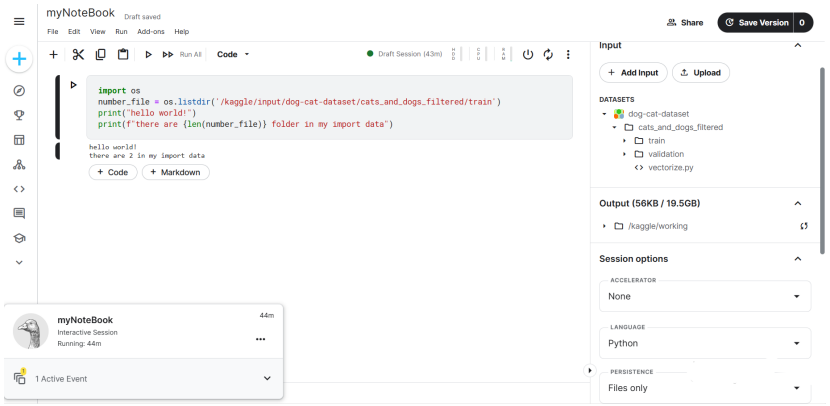
Như đã nói trên, notebooks và dataset là 2 thứ vô cùng quan trọng trong việc xây dựng và huấn luyện mô hình. bạn có thể tạo dataset để sử dụng cho mục đích sau này.



Hình 3: Minh họa việc tạo dataset trên Kaggle

3.2 Kaggle notebook

Bên cạnh dataset, notebooks chính là 1 thứ quan trọng không kém. Một notebook sẽ hoạt động như 1 jupyter notebook thông thường. Trong notebook đó, bạn có thể thêm vào dataset(input bên phải) và sử dụng.



Hình 4: Ví dụ về Kaggle notebook

Khi notebook được chạy bản chất nó vẫn chạy dưới với bộ xử lý cơ bản gần như máy cục bộ. Để sử dụng bộ xử lý GPU của Kaggle cung cấp chọn chuyển giá trị None -> GPU P100/GPU T4x2 ACCELERATOR(Session options). Bên cạnh đó chọn File only trong PERSISTENCE để lưu giữ các file trong Output.

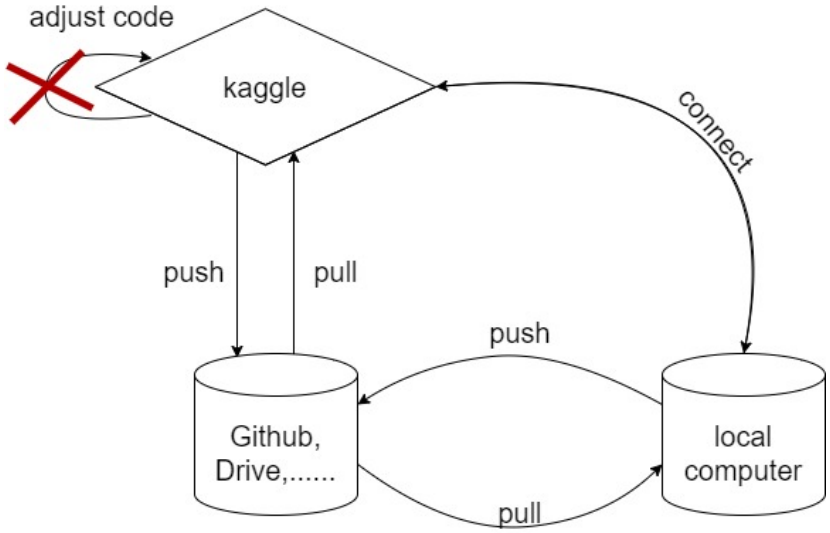
Save Version góc phải hình 4, là chức năng đã được đề cập trong phần Kernels(notebooks) của kiến trúc Kaggle. Save version sẽ giúp người lập trình chạy các đoạn code đã thiệt lập sẵn trong notebook mà không bị gián đoạn trong suốt quá trình huấn luyện mô hình bất kể các yếu tố vật lý như mất kết nối internet, hư hỏng thiết bị. để xem kết quả của phiên bản chạy hãy nhìn vào góc trái dưới hình 4.

4 Kết nối Kaggle với máy tính cục bộ

Như đã nói Kaggle là nền tảng phát triển mạnh giúp các nhà nghiên cứu có thể sử dụng để xây dựng và phát triển mô hình một cách dễ dàng và nhánh chóng. Với bộ xử

lý GPU miễn phí 30h/tuần giúp cho việc chạy thực nghiệm diễn 1 cách nhanh chóng và không bị gián đoạn do bất kỳ tác nhân vật lý nào.

Nhưng điểm hạn chế lớn nhất của Kaggle chính là việc không thể chỉnh sửa được mã nguồn trong Output. Điểm hạn chế này kiến việc chính sửa mã nguồn phải thông qua máy cục bộ đến dịch vụ quản lý mã nguồn như github, drive,.... rồi từ đó chuyển đến Kaggle. Cứ mỗi lần chỉnh sửa code hay debug việc thực hiện diễn ra vô cùng cồng kềnh.



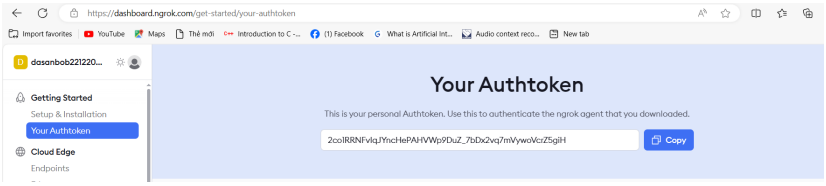
Hình 5: Điểm hạn chế của Kaggle và giải pháp

Để khắc phục được điểm hạn chế này, Kaggle có hổ trợ cho ta kết nối giữa máy tính cục bộ của ta với local thông qua SSH key, ngrok và IDE visual code[2]....

4.1 Thiết lập các cổng, quyền truy cập

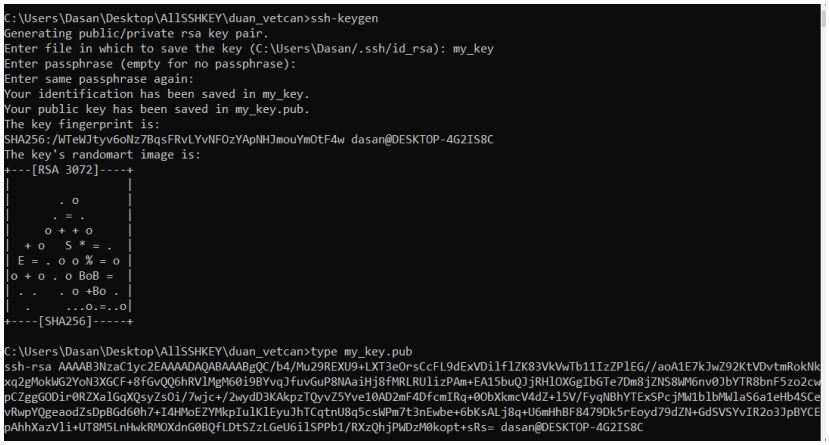
Trước khi nói về chuyện kết nối máy tính cục bộ và Kaggle, phải đẩm bảo một điều rằng bạn đã các đặt đủ các yếu tố: Ngrok token, SSH key

- **Ngrok token**: đóng vai trò như chìa khóa để mở cổng kết nối. Lấy mã Ngrok token bằng các đăng nhập vào địa chỉ này:
  https://dashboard.ngrok.com/login và chọn mục Your Authtoken.



Hình 6: cách lấy giá trị Ngrok token

- **SSH key**: Đây là giá trị quan trọng cấp quyền truy cập local-Kaggle. Giúp ta kết nối giữa Kaggle và máy tính cục bộ của mình. để có giá trị này vào command line sử dụng câu lệnh: ssh-keygen để tạo (password luôn bấm enter) và type [tên ssh-key bạn tạo].

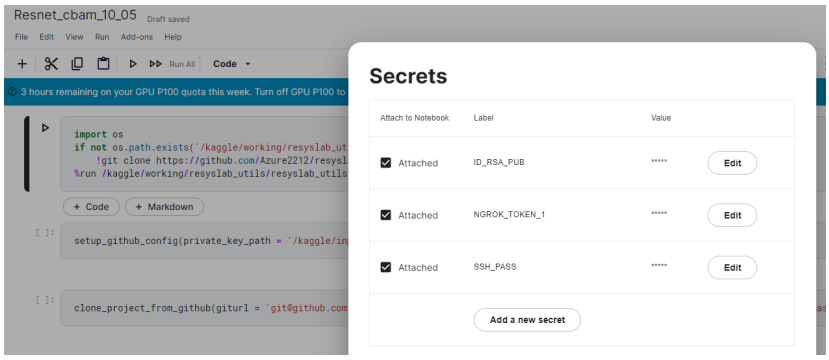


Hình 7: Cách lấy giá trị SSH key

4.2 Các bước kết nối Kaggle-local

Nếu bạn đã đã thực hiện các bước ở phần 4.1, bạn có thể dễ dàng kết nối máy tính cục bộ và Kaggle.

- **cài đặt các cổng trong Kaggle**: Tạo 1 notebook -> Add-ons - > Secrets. Ở đây bạn thêm thuộc tính NGROK_TOKEN_1 mang giá trị ngrok token mà bạn đã lấy ở phần 4.1. Cũng tại đây bạn cũng thêm thuộc tính ID_RSA_PUB mạng giá trị ssh mà bạn cũng đã tạo ở phần 4.1. Và cuối cùng SSH_PASS = 12345. Nếu bạn đã sử cài đặt những thứ này ở các notebook khác thì bạn chỉ cần attach vào mà thôi.



Hình 8: Cách lấy giá trị SSH key

- **Lấy thư viện hổ trợ**: Ở đây bạn sẽ phải clone thư viện resyslab_utils[4] của Trần Anh Tuấn về qua cú sau(đây là câu lệnh bắt buộc bạn phải thực hiện mỗi khi notebook được thực hiện:

```
import os
if not os.path.exists('/Kaggle/working/resyslab_utils'):
    !git clone https://github.com/Azure2212/resyslab_utils.git
%run /Kaggle/working/resyslab_utils/resyslab_utils/clouds/cloud_setup.py
```

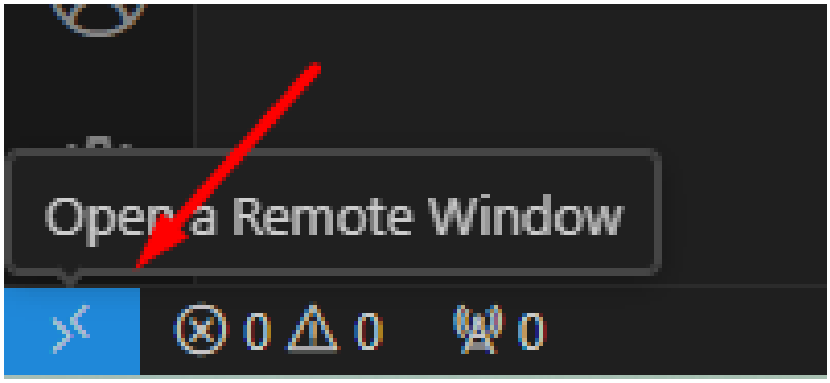
- **Chạy câu lệnh để mở cổng:** connect_vscode()

- **Chọn kết nối từ xa trên vscode:** Ở bước này bạn vào visual code chọn ký hiệu >< góc trái cuối IDE để truy cập cổng kết nối ở hình 9 của bước trước đó.

```
********** SETUP NGROK **********
> Install ngrok...
> Kill ngrok process...
/bin/bash: line 0: kill: `': not a pid or valid job spec
> Binding ports...
> Registry success!
ssh: NgrokTunnel: "tcp://4.tcp.ngrok.io:18475" -> "localhost:22"
vscode: NgrokTunnel: "https://37ea-104-196-27-71.ngrok-free.app" -> "http://localhost:900
0"

---------- Finished ----------
```

Hình 9: Cổng kết nối local - Kaggle



Hình 10: kết nối local - Kaggle

- **Hoàn tất kết nối:** Sau khi bấm >< trong hình 10 sẽ hiện 1 khung cho bạn chọn kết nối. Ở đây bạn chọn Connect to Host(Remote-SSH), với những bạn lần đầu kết nối trên máy chọn cái nào có chữ SSH. Sau đó gõ root@ + port được khoanh trong hình 9, trong ví dụ này gõ root@4.tcp.ngrok.io:18475. Sau đó cứ bấm ok, khúc nào yêu cầu password thì gõ 12345. Khi kết nối thành công góc trái màn hình chổ bạn bấm >< sẽ hiện port kết nối của bạn. Lúc này bạn chỉ cần bấm open folder, gõ tên thư mục cần vào là Kaggle/working thì nó sẽ truy cập thẳng vào thư mục Kaggle/working trên notebook bạn kết nối. Tất cả hoàn tất kết nối.

Hình 11: connect to Host để truy cập local đến Kaggle

5 Kết nối Kaggle với Github

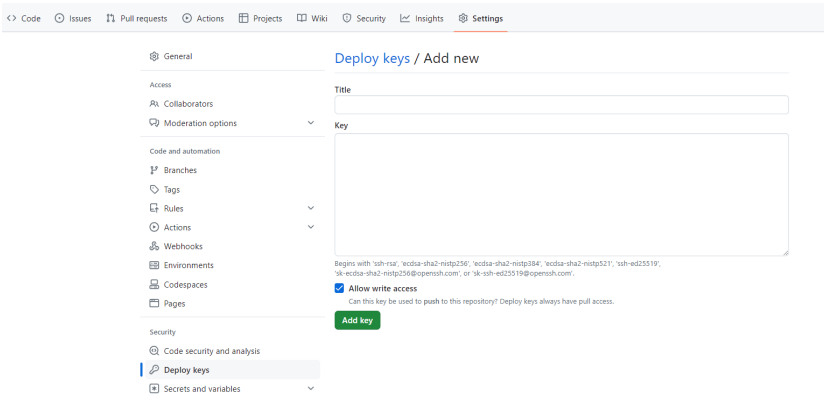
Ở phần trước ta đã nói về cách thức kết nối Kaggle với máy tính cục bộ. Vậy câu hỏi đặt ra làm sao để lưu trữ code để mọi người trong team đều có thể sử dụng trên Kaggle? Để giải quyết vấn đề này phần 5 sẽ nói về việc sử dụng dịch vụ lưu trữ git cụ thể là github[3] trên Kaggle. Các cú pháp để thực hiện git trên Kaggle sẽ sử dụng thư viện resyslab_util của Trần Anh Tuấn đã nhắc ở phần kết nối Kaggle - local trước đó. với các hàm của thư viện được hướng dẫn sử dụng ở:
github.com/Azure2212/resyslab_utils/blob/main/examples/Kaggle_setup.ipynb.

Sẽ luôn có hai dạng thao tác github trên Kaggle. Dạng 1 là lấy dự án của người khác về rồi thực hiện thêm, sửa, xóa rồi push lên lại github, Trong Kaggle chỉ hổ trợ ta clone project bằng link SSH mà để làm được điều này thì yêu cầu chủ dự án bạn clone phải cài

đặt ssh key trong dự án github. Dạng 2 là bạn sẽ đúng ra trực tiếp thiết lập dự án cho các thành viên trong team có thể dễ dàng thao tác được.

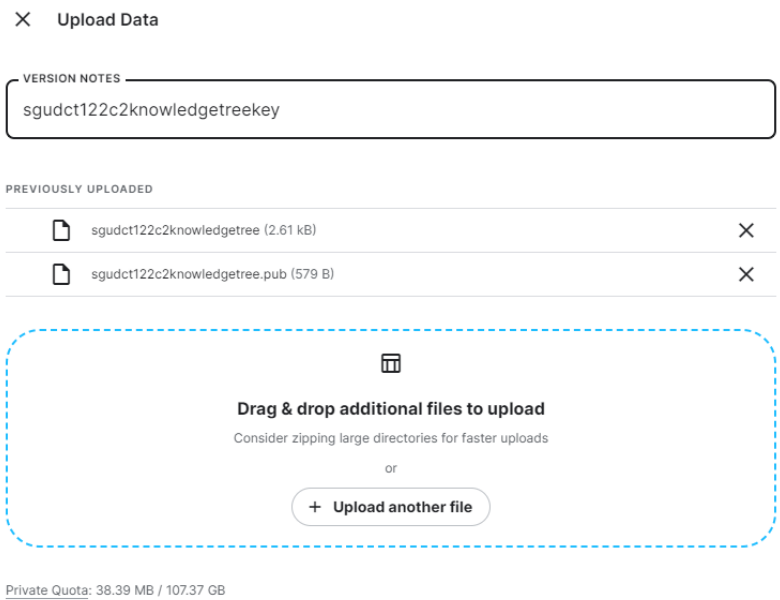
5.1 Làm chủ dự án Github_Kaggle

Để làm được điều này điều tiên quyết: tạo SSH key như đã hướng dẫn phần trên, nắm được các kỹ thuật cơ bản của github. Đầu tiên bạn tạo 1 dự án trên github và thiết lập SSH key cho dự án đó. Bạn vào repository bạn vừa tạo -> setting -> Deploy keys -> add deploy key -> đặt tên và điền key là giá trị key bạn tạo cho dự án (type [tenkey].pub), nhớ chọn Allow right access sau đó Add là xong. Sau đó bạn vào dataset trên Kaggle



Hình 12: Thêm deploy keys vào repository

tạo 1 dataset như đã hướng dẫn ở phần 3.1. Kéo ssh key bạn đã tạo để deploy cho dự án của mình vào(bao gồm cả private và public key) sau đó bấm create. Các bước trên



Hình 13: Thêm dataset chứa quyền truy cập dự án github trên Kaggle

cũng đã hoàn tất quả trình setup dự án Github và tạo quyền truy cập đến dự án đó trên Kaggle. Các thành viên trong team(bao gồm cả bạn) có thể import dataset này vào bất kỳ notebook nào để thực hiện các thao tác trên Kaggle.

5.2 Thao tác dự án Github trên Kaggle

Ở phần 5.1, Ta đã nói về việt cài đặt dự án trên Github và cài đặt quyền truy cập(dataset) vào dự án đó trên Kaggle. Ở phần này sẽ nói về cách Các thành viên trong dự án đó thực hiện điều chỉnh trên Kaggle và cập nhật lên github.

Để Thực hiện được điều này, Hãy tạo 1 notebook. Trong notebook đó bạn import dataset chứa key đã deploy cho dự án mà bạn muốn thao tác trên github.

- **Lấy thư viện hổ trợ**: Ở đây bạn sẽ phải clone thư viện của Trần Anh Tuấn về qua cú sau(đây là câu lệnh bắt buộc bạn phải thực hiện mỗi khi notebook được thực hiện:

  ```
  import os
  if not os.path.exists('/Kaggle/working/resyslab_utils'):
      !git clone https://github.com/Azure2212/resyslab_utils.git
  %run /Kaggle/working/resyslab_utils/resyslab_utils/clouds/cloud_setup
  .py
  ```

- **Cài đặt quyền truy cập**: Ở đây để truy cập hay thao tác để dự án Kaggle bạn phải thiết lập quyền truy cập chi notebook này qua câu lệnh(nhớ thay đổi private_key_path thành private key trên dataset bạn đã import:

  ```
  setup_github_config(private_key_path = '/Kaggle/input/sgudct122c2know
  ledgetreekey/sgudct122c2knowledgetree')
  ```

- **Lấy dự án về notebook Kaggle**: Khi bạn thiết lập key cho phép truy cập vào dự án trên github bạn chỉ đơn giản sử dụng câu lệnh sau để lấy dự án về notebook, đặt lại tên dự án, và chọn nhánh để clone về:

  ```
  clone_project_from_github(folder = '/Kaggle/working/yourproject',
  giturl = 'git@github.com:Azure2212/20032024DCT122C2.git',
  branch = 'yourbranch')
  ```

  Sau khi lấy dự án về notebook bạn hoàn toàn có thể dùng cách kết nối Kaggle - local để lập trình.

- **Cập nhật các thay đổi lên Github**: Sau khi lập trình xong, để cập nhật những thay đổi lên github bạn cần thực hiện lại câu lệnh cài đặt quyền truy cập ở bữa trước. sau đó dùng lệnh: cd /Kaggle/working/yourproject

  Nếu bạn đã đổi tên thì cd vào đúng thư mục của bạn. sử dụng các câu lệnh git cơ bản sau để cập nhật (Nhớ thay đổi lại các thông số phù hợp với dự án của bạn):

  ```
  !git checkout yourbranch
  !git add .
  !git config --global user.email "your mail"
  !git config --global user.name "you name"
  !git commit -m "your comment"
  !git push origin yourbranch
  ```

6 CONCLUSION

Trong bài giới viết này, chúng ta đã khám phá về 1 nền tàng trực tuyến Kaggle và hiểu sâu hơn về cách thực hoạt động của nó, qua đó giúp ta có để dễ dàng sử dụng Kaggle như 1 công cụ để thực hiện các công trình nghiên cứu vì mục tích cá nhân / tập thể 1 cách dễ dàng và vô cùng hiệu quả.

Bài viết Đã chỉ ra các điểm mạnh của Kaggle đó là công động nghiên cứu rộng lớn có thể học hỏi lẫn nhau, cho sử dụng dung lượng GPU 30h/Tuần vô cùng hiệu quả cho việc

nghiên cứu, với số lương dataset có thể thêm và lấy gần như không giới hạn. Nhưng bên cạnh đó như đã nói, Kaggle không để tự chỉnh sửa code được bởi thế bài viết này đã đưa ra cách tiếp cận kết nối Kaggle - local thông qua ngrok token, ssh key, vscode.

Việc lĩnh hội, thành thạo Kaggle cũng như các công cụ hổ trợ Kaggle như github, python hay các kỹ năng quản lý source code, tổ chức hoạt động cũng đòi hỏi phải có 1 khoảng thời gian suy nghĩ và luyên tập. Bài viết này chỉ đề cập đến những thứ cơ bản mà 1 nghiên cứu sinh cần để thực hiện việc nghiên cứu. Ngoài ra còn có rất nhiều kỹ thuật khác như Kaggle - googleDrive,... Người dùng có thể tìm hiểu thêm nếu thấy thật sự cần thiết.

Tài liệu

[1] *Kaggle*, [Online].
Available: `https://www.Kaggle.com/`

[2] *Visual code*, [Online].
Available: `https://code.visualstudio.com/`

[3] *Visual code*, [Online].
Available: `https://github.com/`

[4] *resyslab_ utils*, [Online].
Available: `https://github.com/Azure2212/resyslab_utils`