Sử dụng với Pandas

Tài liệu này là phần giới thiệu nhanh về cách sử dụng bộ dữ liệu với Pandas, với trọng tâm cụ thể là về cách xử lý
tập dữ liệu sử dụng các hàm Pandas và cách chuyển đổi tập dữ liệu sang Pandas hoặc từ Pandas.

Điều này đặc biệt hữu ích vì nó cho phép thực hiện các thao tác nhanh chóng vì bộ dữ liệu sử dụng PyArrow mui xe và PyArrow được tích hợp tốt với Pandas.

Định dạng tập dữ liệu

Theo mặc định, bộ dữ liệu trả về các đối tượng Python thông thường: số nguyên, số float, chuỗi, danh sách, v

Thay vào đó, để nhận Pandas DataFrames hoặc Series, bạn có thể đặt định dạng của tập dữ liệu thành panda using Dataset.with_format():

```
>>> from datasets import Dataset
>>> data = {"col_0": ["a", "b", "c", "d"], "col_1": [0., 0., 1., 1.]}
>>> ds = Dataset.from_dict(data)
>>> ds = ds.with_format("pandas")
>>> ds[0]# pd.DataFrame
   col_0 col_1
0Một0,0
>>> ds[:2]# pd.DataFrame
   col_0 col_1
0Một0,0
1b0,0
>>> ds["data"]# pd.Series
0Một
1b
2c
3d
Tên: col_0, dtype: đối tượng
```

Điều này cũng hoạt động đối với các đối tượng IterableDataset thu được, ví dụ: sử dụng load_dataset(..., streaming=True) :

```
>>> ds = ds.with_format("pandas")
>>> for df in ds.iter(batch_size=2):
...print(df)
...phá vỡ
   col_0 col_1
0Một0,0
1b0,0
```

Xử lý dữ liệu

Các hàm Pandas thường nhanh hơn các hàm python viết tay thông thường và do đó chúng là một lựa chọn tốt để tối ưu hóa việc xử lý dữ liệu. Bạn có thể sử dụng các hàm Pandas để xử lý a dataset in Dataset.map() or Dataset.filter():

```
>>> from datasets import Dataset
>>> data = {"col_0": ["a", "b", "c", "d"], "col_1": [0., 0., 1., 1.]}
>>> ds = Dataset.from_dict(data)
>>> ds = ds.with_format("pandas")
>>> ds = ds.map(lambda df: df.assign(col_2=df.col_1 + 1), batched=True)
>>> ds[:2]
   col_0 col_1 col_2
0Một0,01.0
1b0,01.0
>>> ds = ds.filter(lambda df: df.col_0 == "b", batched=True)
>>> ds[0]
   col_0 col_1 col_2
0b0,01.0
```

We use batched=True because it is faster to process batches of data in Pandas rather than row by row. It's also possible to use batch_size= in map() to set the size of each df .

This also works for IterableDataset.map() and IterableDataset.filter().

Nhập hoặc xuất từ Pandas

To import data from Pandas, you can use Dataset.from_pandas():

```
ds = Dataset.from_pandas(df)
```

And you can use Dataset.to_pandas() to export a Dataset to a Pandas DataFrame:

```
df = Dataset.to_pandas()
```