Lớp bảng

Mỗi đối tượng Dataset được hỗ trợ bởi Bảng PyArrow.
A Table can be loaded from either the disk (memory mapped) or in memory.
Một số loại Bảng có sẵn và tất cả chúng đều kế thừa từ bảng.Table.

Tabledatasets.table.Table

lớp tập dữ liệu.table.Tabledatasets.table.Tablehttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L153[{"name": "table", "val": ": Table"}]

Bao bọc Bảng pyarrow bằng cách sử dụng thành phần.
Đây là lớp cơ sở cho InMemoryTable, MemoryMappedTable và ConcatenationTable.

Nó thực hiện tất cả các thuộc tính/phương thức cơ bản của lớp Bảng pyarrow ngoại trừ
Bảng biến đổi:

cắt, lọc, làm phẳng, Combine_chunks, ép kiểu, add_column,append_column, Remove_column, set_column, và thả .

Việc thực hiện các phương pháp này khác nhau đối với các lớp con.

xác thựcdatasets.table.Table.validatehttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L178[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- full (bool , defaults to False) --
Nếu Đúng , hãy chạy các kiểm tra đắt tiền, nếu không thì chỉ kiểm tra rẻ.0- pa.lib.ArrowInvalid -- if xác thực không thành công pa.lib.ArrowInvalid

Thực hiện kiểm tra xác nhận. Một ngoại lệ được đưa ra nếu xác thực không thành công.

By default only cheap validation checks are run. Pass full=True
for thorough validation checks (potentially O(n)).

Equalsdatasets.table.Table.equalshttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L194[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- other (Table) --
Bảng để so sánh.

- check_metadata bool , defaults to False) --
  Liệu có nên kiểm tra sự bình đẳng của siêu dữ liệu lược đồ hay không.0 bool

Kiểm tra xem nội dung của hai bảng có bằng nhau không.

to_batchesdatasets.table.Table.to_batcheshttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L211[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- max_chunksize (int , defaults to None) --
Kích thước tối đa cho các khối RecordBatch. Các khối riêng lẻ có thể
smaller depending on the chunk layout of individual columns.0 List[pyarrow.RecordBatch]

Convert Table to list of (contiguous) RecordBatch objects.

to_pydictdatasets.table.Table.to_pydicthttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L225[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}] dict

Chuyển đổi Bảng thành dict hoặc OrderedDict .

to_pandasdatasets.table.Table.to_pandashttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L243[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- memory_pool (MemoryPool , defaults to None) --
Mũi tên MemoryPool để sử dụng cho việc phân bổ. Sử dụng bộ nhớ mặc định
hồ bơi không được thông qua.

- strings_to_categorical (bool , defaults to False) --
  Encode string (UTF8) and binary types to pandas.Categorical .
- categories (list , defaults to empty) --
  Danh sách các trường cần được trả về dưới dạng pandas.Categorical . Chỉ một
  áp dụng cho các cấu trúc dữ liệu giống như bảng.
- zero_copy_only (bool , defaults to False) --
  Tăng ArrowException nếu lệnh gọi hàm này yêu cầu sao chép
  dữ liệu cơ bản.
- integer_object_nulls (bool , defaults to False) --
  Truyền số nguyên có giá trị rỗng cho đối tượng.
- date_as_object (bool , defaults to True) --
  Cast dates to objects. If False , convert to datetime64[ns] dtype.
- timestamp_as_object (bool , defaults to False) --

Cast non-nanosecond timestamps (np.datetime64) to objects. This is
hữu ích nếu bạn có dấu thời gian không khớp với ngày bình thường
range of nanosecond timestamps (1678 CE-2262 CE).
If False , all timestamps are converted to datetime64[ns] dtype.

• use_threads (bool , defaults to True) --
Có song song hóa chuyển đổi bằng nhiều luồng hay không.

• deduplicate_objects (bool , defaults to False) --
Không tạo nhiều bản sao đối tượng Python khi tạo, để lưu
về việc sử dụng bộ nhớ. Chuyển đổi sẽ chậm hơn.

• ignore_metadata (bool , defaults to False) --
Nếu Đúng , không sử dụng siêu dữ liệu 'gấu trúc' để xây dựng lại
Chỉ mục DataFrame, nếu có.

• safe (bool , defaults to True) --
Đối với một số kiểu dữ liệu nhất định, cần phải ép kiểu để lưu trữ
data in a pandas DataFrame or Series (e.g. timestamps are always
stored as nanoseconds in pandas). This option controls whether it
có phải là diễn viên an toàn hay không.

• split_blocks (bool , defaults to False) --
Nếu Đúng, hãy tạo một "khối" nội bộ cho mỗi cột khi
tạo pandas.DataFrame từ RecordBatch hoặc Table . Trong khi điều này
có thể tạm thời giảm bộ nhớ lưu ý rằng các hoạt động khác nhau của gấu trúc
có thể kích hoạt "hợp nhất" có thể sử dụng bộ nhớ bong bóng.

• self_destruct (bool , defaults to False) --
THỬ NGHIỆM: Nếu Đúng, hãy cố gắng phân bổ lại Mũi tên ban đầu
bộ nhớ trong khi chuyển đổi đối tượng Arrow thành gấu trúc. Nếu bạn sử dụng
đối tượng sau khi gọi to_pandas bằng tùy chọn này, nó sẽ làm hỏng
chương trình.

• types_mapper (function , defaults to None) --
Một hàm ánh xạ Kiểu dữ liệu pyarrow tới ExtensionDtype của gấu trúc.
Điều này có thể được sử dụng để ghi đè loại gấu trúc mặc định để chuyển đổi
các loại pyarrow tích hợp hoặc không có pandas_metadata trong
Lược đồ bảng. Hàm nhận được một Kiểu dữ liệu pyarrow và được
dự kiến sẽ trả về gấu trúc ExtensionDtype hoặc Không có nếu
chuyển đổi mặc định nên được sử dụng cho loại đó. Nếu bạn có
ánh xạ từ điển, bạn có thể chuyển dict.get dưới dạng function.0 pandas.Series hoặc

pandas.DataFrame pandas.Series hoặc pandas.DataFrame tùy thuộc vào loại đối tượng

Chuyển đổi sang mảng NumPy hoặc DataFrame tương thích với gấu trúc, nếu thích hợp.

to_stringdatasets.table.Table.to_stringhttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L305[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]

fielddatasets.table.Table.fieldhttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L324[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- i (Union[int, str]) --
Chỉ mục hoặc tên của trường cần truy xuất.0 pyarrow.Field

Chọn trường lược đồ theo tên cột hoặc chỉ mục số của nó.

cộtdatasets.table.Table.columnhttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L337[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- i
(Union[int, str]) --
Chỉ mục hoặc tên của cột cần lấy.0 pyarrow.ChunkedArray

Chọn một cột theo tên cột hoặc chỉ mục bằng số.

itercolumnsdatasets.table.Table.itercolumnshttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L350[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}] pyarrow.ChunkedArray

Trình lặp trên tất cả các cột theo thứ tự số của chúng.

Schemadatasets.table.Table.schemahttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L359[] pyarrow.Schema

Lược đồ của bảng và các cột của nó.

cộtdatasets.table.Table.columnshttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L369[] List[pa.ChunkedArray]

Danh sách tất cả các cột theo thứ tự số.

num_columnsdatasets.table.Table.num_columnshttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L379[]int

Số cột trong bảng này.

num_rowsdatasets.table.Table.num_rowshttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L389[]int

Số hàng trong bảng này.

Do định nghĩa của một bảng, tất cả các cột có cùng số lượng
hàng.

Shapedatasets.table.Table.shapehttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L402[] (int, int) Number of rows and number of columns.

Dimensions of the table: (#rows, #columns).

nbytesdatasets.table.Table.nbyteshttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L412[]

Tổng số byte được sử dụng bởi các phần tử của bảng.

InMemoryTabledatasets.table.InMemoryTable

tập dữ liệu lớp.table.InMemoryTabledatasets.table.InMemoryTablehttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L638[{"name": "table", "val": ": Table"}]

Bảng được ghi trong bộ nhớ khi nó được tải vào RAM của người dùng.

Pickling nó sẽ sao chép tất cả dữ liệu bằng bộ nhớ.
Việc triển khai của nó rất đơn giản và sử dụng trực tiếp các phương thức Bảng pyarrow cơ bản.

Điều này khác với bảng MemoryMapped, trong đó bảng Pickling không sao chép tất cả
dữ liệu trong bộ nhớ. Đối với MemoryMapped , thay vào đó, việc giải nén sẽ tải lại bảng từ đĩa.

InMemoryTable phải được sử dụng khi dữ liệu vừa với bộ nhớ, trong khi MemoryMapped được dành riêng c
dữ liệu lớn hơn bộ nhớ hoặc khi bạn muốn chiếm dụng bộ nhớ của ứng dụng
ở mức thấp.

xác thựcdatasets.table.InMemoryTable.validatehttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L178[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- full
(bool , defaults to False) --
Nếu Đúng , hãy chạy các kiểm tra đắt tiền, nếu không thì chỉ kiểm tra rẻ.0- pa.lib.ArrowInvalid -- if

xác thực không thành công pa.lib.ArrowInvalid

Thực hiện kiểm tra xác nhận. Một ngoại lệ được đưa ra nếu xác thực không thành công.

By default only cheap validation checks are run. Pass full=True
for thorough validation checks (potentially O(n)).

Equalsdatasets.table.InMemoryTable.equalshttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L194[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- other (Table) --
Bảng để so sánh.

- check_metadata bool , defaults to False) --
  Liệu có nên kiểm tra sự bình đẳng của siêu dữ liệu lược đồ hay không.0 bool

Kiểm tra xem nội dung của hai bảng có bằng nhau không.

to_batchesdatasets.table.InMemoryTable.to_batcheshttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L211[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- max_chunksize (int , defaults to None) --
Kích thước tối đa cho các khối RecordBatch. Các khối riêng lẻ có thể
smaller depending on the chunk layout of individual columns.0 List[pyarrow.RecordBatch]

Convert Table to list of (contiguous) RecordBatch objects.

to_pydictdatasets.table.InMemoryTable.to_pydicthttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L225[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}] dict

Chuyển đổi Bảng thành dict hoặc OrderedDict .

to_pandasdatasets.table.InMemoryTable.to_pandashttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L243[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- memory_pool (MemoryPool , defaults to None) --
Mũi tên MemoryPool để sử dụng cho việc phân bổ. Sử dụng bộ nhớ mặc định
hồ bơi không được thông qua.

- strings_to_categorical (bool , defaults to False) --
  Encode string (UTF8) and binary types to pandas.Categorical .

• categories (list , defaults to empty) --
  Danh sách các trường cần được trả về dưới dạng pandas.Categorical . Chỉ một
  áp dụng cho các cấu trúc dữ liệu giống như bảng.
• zero_copy_only (bool , defaults to False) --
  Tăng ArrowException nếu lệnh gọi hàm này yêu cầu sao chép
  dữ liệu cơ bản.
• integer_object_nulls (bool , defaults to False) --
  Truyền số nguyên có giá trị rỗng cho đối tượng.
• date_as_object (bool , defaults to True) --
  Cast dates to objects. If False , convert to datetime64[ns] dtype.
• timestamp_as_object (bool , defaults to False) --
  Cast non-nanosecond timestamps (np.datetime64) to objects. This is
  hữu ích nếu bạn có dấu thời gian không khớp với ngày bình thường
  range of nanosecond timestamps (1678 CE-2262 CE).
  If False , all timestamps are converted to datetime64[ns] dtype.
• use_threads (bool , defaults to True) --
  Có song song hóa chuyển đổi bằng nhiều luồng hay không.
• deduplicate_objects (bool , defaults to False) --
  Không tạo nhiều bản sao đối tượng Python khi tạo, để lưu
  về việc sử dụng bộ nhớ. Chuyển đổi sẽ chậm hơn.
• ignore_metadata (bool , defaults to False) --
  Nếu Đúng , không sử dụng siêu dữ liệu 'gấu trúc' để xây dựng lại
  Chỉ mục DataFrame, nếu có.
• safe (bool , defaults to True) --
  Đối với một số loại dữ liệu nhất định, cần phải ép kiểu để lưu trữ
  data in a pandas DataFrame or Series (e.g. timestamps are always
  stored as nanoseconds in pandas). This option controls whether it
  có phải là diễn viên an toàn hay không.
• split_blocks (bool , defaults to False) --
  Nếu Đúng, hãy tạo một "khối" nội bộ cho mỗi cột khi
  tạo pandas.DataFrame từ RecordBatch hoặc Table . Trong khi điều này
  có thể tạm thời giảm bộ nhớ lưu ý rằng các hoạt động khác nhau của gấu trúc
  có thể kích hoạt "hợp nhất" có thể sử dụng bộ nhớ bong bóng.
• self_destruct (bool , defaults to False) --
  THỬ NGHIỆM: Nếu Đúng, hãy cố gắng phân bổ lại Mũi tên ban đầu

bộ nhớ trong khi chuyển đổi đối tượng Arrow thành gấu trúc. Nếu bạn sử dụng đối tượng sau khi gọi to_pandas bằng tùy chọn này, nó sẽ làm hỏng chương trình.

- types_mapper (function , defaults to None) --
  Một hàm ánh xạ Kiểu dữ liệu pyarrow tới ExtensionDtype của gấu trúc. Điều này có thể được sử dụng để ghi đè loại gấu trúc mặc định để chuyển đổi các loại pyarrow tích hợp hoặc không có pandas_metadata trong Lược đồ bảng. Hàm nhận được một Kiểu dữ liệu pyarrow và dự kiến sẽ trả về gấu trúc ExtensionDtype hoặc Không có nếu chuyển đổi mặc định nên được sử dụng cho loại đó. Nếu bạn có ánh xạ từ điển, bạn có thể chuyển dict.get dưới dạng function.0 pandas.Series hoặc pandas.DataFrame pandas.Series hoặc pandas.DataFrame tùy thuộc vào loại đối tượng

Chuyển đổi sang mảng NumPy hoặc DataFrame tương thích với gấu trúc, nếu thích hợp.

to_stringdatasets.table.InMemoryTable.to_stringhttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L305[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]

fielddatasets.table.InMemoryTable.fieldhttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L324[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- i
(Union[int, str]) --
Chỉ mục hoặc tên của trường cần lấy.0 pyarrow.Field

Chọn trường lược đồ theo tên cột hoặc chỉ mục số của nó.

cộtdatasets.table.InMemoryTable.columnhttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L337[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- i
(Union[int, str]) --
Chỉ mục hoặc tên của cột cần lấy.0 pyarrow.ChunkedArray

Chọn một cột theo tên cột hoặc chỉ mục bằng số.

itercolumnsdatasets.table.InMemoryTable.itercolumnshttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L350[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}] pyarrow.ChunkedArray

Trình lặp trên tất cả các cột theo thứ tự số của chúng.

Schemadatasets.table.InMemoryTable.schemahttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L359[] pyarrow.Schema

Lược đồ của bảng và các cột của nó.

cộtdatasets.table.InMemoryTable.columnshttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L369[] List[pa.ChunkedArray]

Danh sách tất cả các cột theo thứ tự số.

num_columnsdatasets.table.InMemoryTable.num_columnshttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L379[]int

Số cột trong bảng này.

num_rowsdatasets.table.InMemoryTable.num_rowshttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L389[]int

Số hàng trong bảng này.

Do định nghĩa của một bảng, tất cả các cột có cùng số lượng
hàng.

Shapedatasets.table.InMemoryTable.shapehttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L402[] (int, int) Number of rows and number of columns.

Dimensions of the table: (#rows, #columns).

nbytesdatasets.table.InMemoryTable.nbyteshttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L412[]

Tổng số byte được sử dụng bởi các phần tử của bảng.

cột_namesdatasets.table.InMemoryTable.column_nameshttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L419[]

Tên các cột của bảng.

slicedatasets.table.InMemoryTable.slicehttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L793[{"name": "offset", "val": " = 0"}, {"name": "length", "val": " = None"}]- offset (int , defaults to 0) --

Offset từ đầu bảng đến lát cắt.

- length (int , defaults to None) --
  Length of slice (default is until end of table starting from
  offset).0 datasets.table.Table

Tính toán phần không sao chép của Bảng này.

filterdatasets.table.InMemoryTable.filterhttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L810[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]

Chọn các bản ghi từ một Bảng. Xem pyarrow.compute.filter để biết cách sử dụng đầy đủ.

Flattendatasets.table.InMemoryTable.flattenhttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L816[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- memory_pool (MemoryPool , defaults to None) --
Để phân bổ bộ nhớ, nếu được yêu cầu, nếu không, hãy sử dụng bộ dữ liệu pool.0 mặc định.table.Table

Làm phẳng cái bàn này. Mỗi cột có kiểu cấu trúc được làm phẳng
thành một cột cho mỗi trường cấu trúc. Các cột khác không thay đổi.

Combine_chunksdatasets.table.InMemoryTable.combine_chunkshttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L830[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- memory_pool (MemoryPool , defaults to None) --
Để phân bổ bộ nhớ, nếu được yêu cầu, nếu không, hãy sử dụng bộ dữ liệu pool.0 mặc định.table.Table

Tạo một bảng mới bằng cách kết hợp các khối mà bảng này có.

Tất cả các phần cơ bản trong ChunkedArray của mỗi cột đều
nối thành 0 hoặc một đoạn.

castdatasets.table.InMemoryTable.casthttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L846[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- target_schema (Schema) --
Lược đồ để truyền tới, tên và thứ tự của các trường phải khớp nhau.

- safe (bool , defaults to True) --
  Kiểm tra tình trạng tràn hoặc các bộ dữ liệu chuyển đổi không an toàn khác.table.Table

Truyền các giá trị bảng sang một lược đồ khác.

thay thế_schema_metadatadatasets.table.InMemoryTable.replace_schema_metadatahttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L861[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- metadata (dict , defaults to None)
--0 bộ dữ liệu.table.Table nông_copy

THỬ NGHIỆM: Tạo bản sao nông của bảng bằng cách thay thế lược đồ
key-value metadata with the indicated new metadata (which may be None ,
which deletes any existing metadata).

add_columndatasets.table.InMemoryTable.add_columnhttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L875[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- i (int) --
Chỉ mục để đặt cột tại.

- field_ (Union[str, pyarrow.Field]) --
  Nếu một chuỗi được truyền thì loại được suy ra từ cột
  dữ liệu.
- column (Union[pyarrow.Array, List[pyarrow.Array]]) --
  Cột data.0datadatas.table.Table Bảng mới đã thêm cột đã truyền.

Thêm cột vào Bảng tại vị trí.

Một bảng mới được trả về với cột được thêm vào, bảng gốc
đối tượng được giữ nguyên không thay đổi.

append_columndatasets.table.InMemoryTable.append_columnhttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L896[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- field_ (Union[str, pyarrow.Field]) --
Nếu một chuỗi được truyền thì loại được suy ra từ cột
dữ liệu.

- column (Union[pyarrow.Array, List[pyarrow.Array]]) --
  Cột data.0datadatas.table.Table Bảng mới đã thêm cột đã truyền.

Nối cột vào cuối cột.

Remove_columndatasets.table.InMemoryTable.remove_columnhttps://github.com/huggingface/

datasets/blob/4.2.0/src/datasets/table.py#L913[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- i (int) --
Chỉ mục của cột cần loại bỏ.0datadatas.table.Table Bảng mới không có cột.

Tạo Bảng mới với cột được chỉ định đã bị xóa.

set_columndatasets.table.InMemoryTable.set_columnhttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L927[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- i (int) --
Chỉ mục để đặt cột tại.

- field_ (Union[str, pyarrow.Field]) --
  Nếu một chuỗi được truyền thì loại được suy ra từ cột
  dữ liệu.
- column (Union[pyarrow.Array, List[pyarrow.Array]]) --
  Cột data.0datadatas.table.Table Bảng mới với tập hợp cột đã truyền.

Thay thế cột trong Bảng tại vị trí.

đổi tên_columnsdatasets.table.InMemoryTable.rename_columnshttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L946[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]

Tạo bảng mới với các cột được đổi tên thành tên được cung cấp.

selectdatasets.table.InMemoryTable.selecthttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L969[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- columns (Union[List[str], List[int]]) --
Tên cột hoặc chỉ số nguyên cho bảng select.0datasets.table.TableNew với
các cột được chỉ định và siêu dữ liệu được bảo tồn.

Chọn các cột của bảng.

Trả về một bảng mới với các cột được chỉ định và siêu dữ liệu được giữ nguyên.

dropdatasets.table.InMemoryTable.drophttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L952[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- columns (List[str]) --
Danh sách tên trường tham chiếu các cột hiện có.0 tập dữ liệu.table.Bảng Bảng mới không có

cột.- KeyError -- : nếu bất kỳ tên cột nào được truyền không tồn tại. Lỗi phím

Bỏ một hoặc nhiều cột và trả về một bảng mới.

from_filedatasets.table.InMemoryTable.from_filehttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L653[{"name": "filename", "val": ": str"}]

from_bufferdatasets.table.InMemoryTable.from_bufferhttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L658[{"name": "buffer", "val": ": Buffer"}]

from_pandasdatasets.table.InMemoryTable.from_pandashttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L663[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- df (pandas.DataFrame) --

- schema (pyarrow.Schema , optional) --
  Lược đồ dự kiến của Bảng Mũi tên. Điều này có thể được sử dụng để
  cho biết loại cột nếu chúng ta không thể tự động suy ra nó.
  Nếu được thông qua, đầu ra sẽ có chính xác lược đồ này. Cột
  được chỉ định trong lược đồ không tìm thấy trong các cột DataFrame
  hoặc chỉ mục của nó sẽ gây ra lỗi. Cột hoặc chỉ mục bổ sung
  các mức trong DataFrame không được chỉ định trong lược đồ sẽ
  bị bỏ qua.
- preserve_index (bool , optional) --
  Có lưu trữ chỉ mục dưới dạng cột bổ sung trong kết quả hay không
  Bàn . Giá trị mặc định của Không sẽ lưu trữ chỉ mục dưới dạng cột,
  ngoại trừ RangeIndex chỉ được lưu trữ dưới dạng siêu dữ liệu. Sử dụng
  preserve_index=True to force it to be stored as a column.
- nthreads (int , defaults to None (may use up to system CPU count threads)) --
  Nếu lớn hơn 1, hãy chuyển đổi song song các cột thành Mũi tên bằng cách sử dụng
  số lượng chủ đề được chỉ định.
- columns (List[str] , optional) --
  Danh sách cột cần chuyển đổi Nếu Không có, hãy sử dụng tất cả các cột.
- safe (bool , defaults to True) --
  Kiểm tra tình trạng tràn hoặc các chuyển đổi không an toàn khác, 0 bộ dữ liệu.table.Table

Chuyển đổi pandas.DataFrame thành Bảng mũi tên.

Các loại cột trong Bảng Mũi tên kết quả được suy ra từ

dtypes của pandas.Series trong DataFrame. Trong trường hợp phi vật thể
Series, dtype NumPy được dịch sang dạng tương đương với Arrow của nó. trong
trường hợp của đối tượng, chúng ta cần đoán kiểu dữ liệu bằng cách nhìn vào
Các đối tượng Python trong Series này.

Xin lưu ý rằng Chuỗi đối tượng dtype không mang đủ
thông tin luôn dẫn đến một loại Mũi tên có ý nghĩa. Trong trường hợp đó
chúng ta không thể suy ra một loại, ví dụ: vì DataFrame có độ dài 0 hoặc
Sê-ri chỉ chứa các đối tượng Không có/nan, loại được đặt thành
vô giá trị. Hành vi này có thể tránh được bằng cách xây dựng một lược đồ rõ ràng
và chuyển nó đến chức năng này.

Ví dụ:

```
>>> import pandas as pd
>>> import pyarrow as pa
>>> df = pd.DataFrame({
...'int': [1, 2],
...'str': ['a', 'b']
... })
>>> pa.Table.from_pandas(df)
<pyarrow.lib.Table object at 0x7f05d1fb1b40>
```

from_arraysdatasets.table.InMemoryTable.from_arrayshttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L721[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- arrays (List[Union[pyarrow.Array, pyarrow.ChunkedArray]]) --
Các mảng có độ dài bằng nhau sẽ tạo thành bảng.

- names (List[str] , optional) --
  Tên cho các cột của bảng. Nếu không được thông qua, lược đồ phải được thông qua.
- schema (Schema , defaults to None) --
  Lược đồ cho bảng đã tạo. Nếu không được thông qua, tên phải được thông qua.
- metadata (Union[dict, Mapping] , defaults to None) --
  Optional metadata for the schema (if inferred).0 datasets.table.Table

Xây dựng một bảng từ mảng mũi tên.

from_pydictdatasets.table.InMemoryTable.from_pydicthttps://github.com/huggingface/datasets/

blob/4.2.0/src/datasets/table.py#L741[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- mapping (Union[dict, Mapping]) --
Ánh xạ các chuỗi tới danh sách Mảng hoặc Python.

- schema (Schema , defaults to None) --
  Nếu không được thông qua, sẽ được suy ra từ các giá trị Ánh xạ
- metadata (Union[dict, Mapping] , defaults to None) --
  Optional metadata for the schema (if inferred).0 datasets.table.Table

Xây dựng Bảng từ mảng hoặc cột Mũi tên.

from_batchesdatasets.table.InMemoryTable.from_batcheshttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L777[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- batches (Union[Sequence[pyarrow.RecordBatch], Iterator[pyarrow.RecordBatch]])
None
Trình tự RecordBatch cần chuyển đổi, tất cả các lược đồ phải bằng nhau.

- schema (Schema , defaults to None) --
  Nếu không được thông qua, sẽ được suy ra từ tập dữ liệu RecordBatch .0 đầu tiên.table.Table

Xây dựng Bảng từ một chuỗi hoặc trình lặp của Arrow RecordBatches.

MemoryMappedTabledatasets.table.MemoryMappbảng ed

lớp học
bộ dữ liệu.table.MemoryMappedTabledatasets.table.MemoryMappedTablehttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L989[{"name": "table", "val": ": Table"}, {"name": "path", "val": ": str"}, {"name": "replays", "val": ": typing.Optional[list[tuple[str, tuple, dict]]] = None"}]

Bảng được gọi là ánh xạ bộ nhớ khi nó không sử dụng RAM của người dùng mà tải dữ liệu thay vào đó là từ đĩa.

Pickling nó không sao chép dữ liệu vào bộ nhớ.
Thay vào đó, chỉ đường dẫn đến tệp mũi tên được ánh xạ bộ nhớ được chọn, cũng như danh sách of chuyển đổi thành "phát lại" khi tải lại bảng từ đĩa.

Việc triển khai nó yêu cầu lưu trữ lịch sử của tất cả các phép biến đổi đã được áp dụng
vào Bảng pyarrow bên dưới, để chúng có thể được "phát lại" khi tải lại Bảng
từ đĩa.

Điều này khác với bảng InMemoryTable, trong đó bảng tẩy sẽ sao chép tất cả
dữ liệu trong bộ nhớ.

InMemoryTable phải được sử dụng khi dữ liệu vừa với bộ nhớ, trong khi MemoryMapped được dành riêng c
dữ liệu lớn hơn bộ nhớ hoặc khi bạn muốn chiếm dụng bộ nhớ của ứng dụng
ở mức thấp.

xác thựcdatasets.table.MemoryMappedTable.validatehttps://github.com/huggingface/datasets/
blob/4.2.0/src/datasets/table.py#L178[{"name": "*args", "val": ""}, {"name": "**kwargs", "val":
""}]- full (bool , defaults to False) --
Nếu Đúng , hãy chạy các kiểm tra đắt tiền, nếu không thì chỉ kiểm tra rẻ.0- pa.lib.ArrowInvalid -- if
xác thực không thành công pa.lib.ArrowInvalid

Thực hiện kiểm tra xác nhận. Một ngoại lệ được đưa ra nếu xác thực không thành công.

By default only cheap validation checks are run. Pass full=True
for thorough validation checks (potentially O(n)).

Equalsdatasets.table.MemoryMappedTable.equalshttps://github.com/huggingface/datasets/
blob/4.2.0/src/datasets/table.py#L194[{"name": "*args", "val": ""}, {"name": "**kwargs", "val":
""}]- other (Table) --
Bảng để so sánh.

- check_metadata bool , defaults to False) --
  Liệu có nên kiểm tra sự bình đẳng của siêu dữ liệu lược đồ hay không.0 bool

Kiểm tra xem nội dung của hai bảng có bằng nhau không.

to_batchesdatasets.table.MemoryMappedTable.to_batcheshttps://github.com/huggingface/
datasets/blob/4.2.0/src/datasets/table.py#L211[{"name": "*args", "val": ""}, {"name": "**kwargs",
"val": ""}]- max_chunksize (int , defaults to None) --
Kích thước tối đa cho các khối RecordBatch. Các khối riêng lẻ có thể
smaller depending on the chunk layout of individual columns.0 List[pyarrow.RecordBatch]

Convert Table to list of (contiguous) RecordBatch objects.

to_pydictdatasets.table.MemoryMappedTable.to_pydicthttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L225[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}] dict

Chuyển đổi Bảng thành dict hoặc OrderedDict .

to_pandasdatasets.table.MemoryMappedTable.to_pandashttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L243[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- memory_pool (MemoryPool , defaults to None) --
Mũi tên MemoryPool để sử dụng cho việc phân bổ. Sử dụng bộ nhớ mặc định
hồ bơi không được thông qua.

- strings_to_categorical (bool , defaults to False) --
  Encode string (UTF8) and binary types to pandas.Categorical .
- categories (list , defaults to empty) --
  Danh sách các trường cần được trả về dưới dạng pandas.Categorical . Chỉ một
  áp dụng cho các cấu trúc dữ liệu giống như bảng.
- zero_copy_only (bool , defaults to False) --
  Tăng ArrowException nếu lệnh gọi hàm này yêu cầu sao chép
  dữ liệu cơ bản.
- integer_object_nulls (bool , defaults to False) --
  Truyền số nguyên có giá trị rỗng cho đối tượng.
- date_as_object (bool , defaults to True) --
  Cast dates to objects. If False , convert to datetime64[ns] dtype.
- timestamp_as_object (bool , defaults to False) --
  Cast non-nanosecond timestamps (np.datetime64) to objects. This is
  hữu ích nếu bạn có dấu thời gian không khớp với ngày bình thường
  range of nanosecond timestamps (1678 CE-2262 CE).
  If False , all timestamps are converted to datetime64[ns] dtype.
- use_threads (bool , defaults to True) --
  Có song song hóa chuyển đổi bằng nhiều luồng hay không.
- deduplicate_objects (bool , defaults to False) --
  Không tạo nhiều bản sao đối tượng Python khi tạo, để lưu
  về việc sử dụng bộ nhớ. Chuyển đổi sẽ chậm hơn.
- ignore_metadata (bool , defaults to False) --
  Nếu Đúng , không sử dụng siêu dữ liệu 'gấu trúc' để xây dựng lại

Chỉ mục DataFrame, nếu có.

- safe (bool , defaults to True) --
  Đối với một số loại dữ liệu nhất định, cần phải ép kiểu để lưu trữ
  data in a pandas DataFrame or Series (e.g. timestamps are always
  stored as nanoseconds in pandas). This option controls whether it
  có phải là diễn viên an toàn hay không.
- split_blocks (bool , defaults to False) --
  Nếu Đúng, hãy tạo một "khối" nội bộ cho mỗi cột khi
  tạo pandas.DataFrame từ RecordBatch hoặc Table . Trong khi điều này
  có thể tạm thời giảm bộ nhớ lưu ý rằng các hoạt động khác nhau của gấu trúc
  có thể kích hoạt "hợp nhất" có thể sử dụng bộ nhớ bong bóng.
- self_destruct (bool , defaults to False) --
  THỬ NGHIỆM: Nếu Đúng, hãy cố gắng phân bổ lại Mũi tên ban đầu
  bộ nhớ trong khi chuyển đổi đối tượng Arrow thành gấu trúc. Nếu bạn sử dụng
  đối tượng sau khi gọi to_pandas bằng tùy chọn này, nó sẽ làm hỏng
  chương trình.
- types_mapper (function , defaults to None) --
  Một hàm ánh xạ Kiểu dữ liệu pyarrow tới ExtensionDtype của gấu trúc.
  Điều này có thể được sử dụng để ghi đè loại gấu trúc mặc định để chuyển đổi
  các loại pyarrow tích hợp hoặc không có pandas_metadata trong
  Lược đồ bảng. Hàm nhận được một Kiểu dữ liệu pyarrow và
  dự kiến sẽ trả về gấu trúc ExtensionDtype hoặc Không có nếu
  chuyển đổi mặc định nên được sử dụng cho loại đó. Nếu bạn có
  ánh xạ từ điển, bạn có thể chuyển dict.get dưới dạng function.0 pandas.Series hoặc
  pandas.DataFrame pandas.Series hoặc pandas.DataFrame tùy thuộc vào loại đối tượng

Chuyển đổi sang mảng NumPy hoặc DataFrame tương thích với gấu trúc, nếu thích hợp.

to_stringdatasets.table.MemoryMappedTable.to_stringhttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L305[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]

fielddatasets.table.MemoryMappedTable.fieldhttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L324[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- i (Union[int, str]) --
Chỉ mục hoặc tên của trường cần lấy.0 pyarrow.Field

Chọn trường lược đồ theo tên cột hoặc chỉ mục số của nó.

cộtdatasets.table.MemoryMappedTable.columnhttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L337[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- i (Union[int, str]) --
Chỉ mục hoặc tên của cột cần lấy.0 pyarrow.ChunkedArray

Chọn một cột theo tên cột hoặc chỉ mục bằng số.

itercolumnsdatasets.table.MemoryMappedTable.itercolumnshttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L350[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}] pyarrow.ChunkedArray

Trình lặp trên tất cả các cột theo thứ tự số của chúng.

Schemadatasets.table.MemoryMappedTable.schemahttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L359[] pyarrow.Schema

Lược đồ của bảng và các cột của nó.

cộtdatasets.table.MemoryMappedTable.columnshttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L369[] List[pa.ChunkedArray]

Danh sách tất cả các cột theo thứ tự số.

num_columnsdatasets.table.MemoryMappedTable.num_columnshttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L379[]int

Số cột trong bảng này.

num_rowsdatasets.table.MemoryMappedTable.num_rowshttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L389[]int

Số hàng trong bảng này.

Do định nghĩa của một bảng, tất cả các cột có cùng số lượng
hàng.

Shapedatasets.table.MemoryMappedTable.shapehttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L402[] (int, int) Number of rows and number of columns.

Dimensions of the table: (#rows, #columns).

nbytesdatasets.table.MemoryMappedTable.nbyteshttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L412[]

Tổng số byte được sử dụng bởi các phần tử của bảng.

cột_namesdatasets.table.MemoryMappedTable.column_nameshttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L419[]

Tên các cột của bảng.

slicedatasets.table.MemoryMappedTable.slicehttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L1048[{"name": "offset", "val": " = 0"}, {"name": "length", "val": " = None"}]- offset (int , defaults to 0) --
Offset từ đầu bảng đến lát cắt.

- length (int , defaults to None) --
  Length of slice (default is until end of table starting from
  offset).0 datasets.table.Table

Tính toán phần không sao chép của Bảng này.

filterdatasets.table.MemoryMappedTable.filterhttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L1067[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]

Chọn các bản ghi từ một Bảng. Xem pyarrow.compute.filter để biết cách sử dụng đầy đủ.

Flattendatasets.table.MemoryMappedTable.flattenhttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L1075[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- memory_pool (MemoryPool , defaults to None) --
Để phân bổ bộ nhớ, nếu được yêu cầu, nếu không, hãy sử dụng bộ dữ liệu pool.0 mặc định.table.Table

Làm phẳng cái bàn này. Mỗi cột có kiểu cấu trúc được làm phẳng
thành một cột cho mỗi trường cấu trúc. Các cột khác không thay đổi.

Combine_chunksdatasets.table.MemoryMappedTable.combine_chunkshttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L1091[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- memory_pool (MemoryPool , defaults to None) --
Để phân bổ bộ nhớ, nếu được yêu cầu, nếu không, hãy sử dụng bộ dữ liệu pool.0 mặc định.table.Table

Tạo một bảng mới bằng cách kết hợp các khối mà bảng này có.

Tất cả các phần cơ bản trong ChunkedArray của mỗi cột đều
nối thành 0 hoặc một đoạn.

castdatasets.table.MemoryMappedTable.casthttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L1109[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- target_schema (Schema) --
Lược đồ để truyền tới, tên và thứ tự của các trường phải khớp nhau.

- safe (bool , defaults to True) --
  Kiểm tra tình trạng tràn hoặc các bộ dữ liệu chuyển đổi không an toàn khác.table.Table

Truyền giá trị bảng sang lược đồ khác

thay thế_schema_metadatadatasets.table.MemoryMappedTable.replace_schema_metadatahttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L1126[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- metadata (dict , defaults to None)
--0 bộ dữ liệu.table.Table nông_copy

THỬ NGHIỆM: Tạo bản sao nông của bảng bằng cách thay thế lược đồ
key-value metadata with the indicated new metadata (which may be None,
sẽ xóa mọi siêu dữ liệu hiện có.

add_columndatasets.table.MemoryMappedTable.add_columnhttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L1142[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- i (int) --
Chỉ mục để đặt cột tại.

- field_ (Union[str, pyarrow.Field]) --
  Nếu một chuỗi được truyền thì loại được suy ra từ cột
  dữ liệu.
- column (Union[pyarrow.Array, List[pyarrow.Array]]) --
  Cột data.0datadatas.table.Table Bảng mới đã thêm cột đã truyền.

Thêm cột vào Bảng tại vị trí.

Một bảng mới được trả về với cột được thêm vào, bảng gốc
đối tượng được giữ nguyên không thay đổi.

append_columndatasets.table.MemoryMappedTable.append_columnhttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L1165[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- field_ (Union[str, pyarrow.Field]) --
Nếu một chuỗi được truyền thì loại được suy ra từ cột
dữ liệu.

- column (Union[pyarrow.Array, List[pyarrow.Array]]) --
  Cột data.0datadatas.table.Table Bảng mới đã thêm cột đã truyền.

Nối cột vào cuối cột.

Remove_columndatasets.table.MemoryMappedTable.remove_columnhttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L1184[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- i (int) --
Chỉ mục của cột cần loại bỏ.0datadatas.table.Table Bảng mới không có cột.

Tạo Bảng mới với cột được chỉ định đã bị xóa.

set_columndatasets.table.MemoryMappedTable.set_columnhttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L1200[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- i (int) --
Chỉ mục để đặt cột tại.

- field_ (Union[str, pyarrow.Field]) --
  Nếu một chuỗi được truyền thì loại được suy ra từ cột
  dữ liệu.
- column (Union[pyarrow.Array, List[pyarrow.Array]]) --
  Cột data.0datadatas.table.Table Bảng mới với tập hợp cột đã truyền.

Thay thế cột trong Bảng tại vị trí.

đổi tên_columnsdatasets.table.MemoryMappedTable.rename_columnshttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L1221[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]

Tạo bảng mới với các cột được đổi tên thành tên được cung cấp.

selectdatasets.table.MemoryMappedTable.selecthttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L1248[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]-

columns (Union[List[str], List[int]]) --
Tên cột hoặc chỉ số nguyên cho bảng select.0datasets.table.TableNew với
các cột được chỉ định và siêu dữ liệu được bảo tồn.

Chọn các cột của bảng.

Trả về một bảng mới với các cột được chỉ định và siêu dữ liệu được giữ nguyên.

dropdatasets.table.MemoryMappedTable.drophttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L1229[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]-
columns (List[str]) --
Danh sách tên trường tham chiếu các cột hiện có.0 tập dữ liệu.table.Bảng Bảng mới không có cột.- KeyError -- : nếu bất kỳ tên cột nào được truyền không tồn tại. Lỗi phím

Bỏ một hoặc nhiều cột và trả về một bảng mới.

from_filedatasets.table.MemoryMappedTable.from_filehttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L1015[{"name": "filename", "val": ": str"}, {"name": "replays", "val": " = None"}]

ConcatenationTabledatasets.table.ConcatenationTcó thể

tập dữ liệu lớp.table.ConcatenationTabledatasets.table.ConcatenationTablehttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L1273[{"name": "table", "val": ": Table"}, {"name": "blocks", "val": ": list"}]

Bảng này xuất phát từ sự ghép nối của một số bảng được gọi là khối.
It enables concatenation on both axis 0 (append rows) and axis 1 (append columns).

Các bảng cơ bản được gọi là "khối" và có thể là InMemoryTable
hoặc các đối tượng MemoryMappedTable.
Điều này cho phép kết hợp các bảng đến từ bộ nhớ hoặc được ánh xạ bộ nhớ.
Khi một ConcatenationTable được chọn, thì mỗi khối sẽ được chọn:

- các đối tượng InMemoryTable được chọn lọc bằng cách sao chép tất cả dữ liệu trong bộ nhớ.
- các đối tượng MemoryMappedTable được chọn mà không cần sao chép dữ liệu vào bộ nhớ.

Thay vào đó, chỉ đường dẫn đến tệp mũi tên được ánh xạ bộ nhớ được chọn, cũng như danh sách chuyển đổi thành "phát lại" khi tải lại bảng từ đĩa.

Việc thực hiện nó yêu cầu lưu trữ từng khối riêng biệt.
Thuộc tính khối lưu trữ danh sách các khối.
The first axis concatenates the tables along the axis 0 (it appends rows),
while the second axis concatenates tables along the axis 1 (it appends columns).

Nếu một số cột bị thiếu khi nối trên trục 0, chúng sẽ chứa giá trị null.
This is done using pyarrow.concat_tables(tables, promote=True) .

Bạn có thể truy cập bảng kết hợp đầy đủ bằng cách truy cập thuộc tính ConcatenationTable.table, và các khối bằng cách truy cập thuộc tính ConcatenationTable.blocks.

xác thựcdatasets.table.ConcatenationTable.validatehttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L178[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- full (bool , defaults to False) --
Nếu Đúng , hãy chạy các kiểm tra đắt tiền, nếu không thì chỉ kiểm tra rẻ.0- pa.lib.ArrowInvalid -- if xác thực không thành công pa.lib.ArrowInvalid

Thực hiện kiểm tra xác nhận. Một ngoại lệ được đưa ra nếu xác thực không thành công.

By default only cheap validation checks are run. Pass full=True
for thorough validation checks (potentially O(n)).

Equalsdatasets.table.ConcatenationTable.equalshttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L194[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- other (Table) --
Bảng để so sánh.

- check_metadata bool , defaults to False) --
  Liệu có nên kiểm tra sự bình đẳng của siêu dữ liệu lược đồ hay không.0 bool

Kiểm tra xem nội dung của hai bảng có bằng nhau không.

to_batchesdatasets.table.ConcatenationTable.to_batcheshttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L211[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- max_chunksize (int , defaults to None) --
Kích thước tối đa cho các khối RecordBatch. Các khối riêng lẻ có thể

smaller depending on the chunk layout of individual columns.0 List[pyarrow.RecordBatch]

Convert Table to list of (contiguous) RecordBatch objects.

to_pydictdatasets.table.ConcatenationTable.to_pydicthttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L225[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}] dict

Chuyển đổi Bảng thành dict hoặc OrderedDict .

to_pandasdatasets.table.ConcatenationTable.to_pandashttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L243[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- memory_pool (MemoryPool , defaults to None) --
Mũi tên MemoryPool để sử dụng cho việc phân bổ. Sử dụng bộ nhớ mặc định
hồ bơi không được thông qua.

- strings_to_categorical (bool , defaults to False) --
  Encode string (UTF8) and binary types to pandas.Categorical .
- categories (list , defaults to empty) --
  Danh sách các trường cần được trả về dưới dạng pandas.Categorical . Chỉ một
  áp dụng cho các cấu trúc dữ liệu giống như bảng.
- zero_copy_only (bool , defaults to False) --
  Tăng ArrowException nếu lệnh gọi hàm này yêu cầu sao chép
  dữ liệu cơ bản.
- integer_object_nulls (bool , defaults to False) --
  Truyền số nguyên có giá trị rỗng cho đối tượng.
- date_as_object (bool , defaults to True) --
  Cast dates to objects. If False , convert to datetime64[ns] dtype.
- timestamp_as_object (bool , defaults to False) --
  Cast non-nanosecond timestamps (np.datetime64) to objects. This is
  hữu ích nếu bạn có dấu thời gian không khớp với ngày bình thường
  range of nanosecond timestamps (1678 CE-2262 CE).
  If False , all timestamps are converted to datetime64[ns] dtype.
- use_threads (bool , defaults to True) --
  Có song song hóa chuyển đổi bằng nhiều luồng hay không.
- deduplicate_objects (bool , defaults to False) --
  Không tạo nhiều bản sao đối tượng Python khi tạo, để lưu

về việc sử dụng bộ nhớ. Chuyển đổi sẽ chậm hơn.

- ignore_metadata (bool , defaults to False) --
  Nếu Đúng , không sử dụng siêu dữ liệu 'gấu trúc' để xây dựng lại
  Chỉ mục DataFrame, nếu có.
- safe (bool , defaults to True) --
  Đối với một số kiểu dữ liệu nhất định, cần phải ép kiểu để lưu trữ
  data in a pandas DataFrame or Series (e.g. timestamps are always
  stored as nanoseconds in pandas). This option controls whether it
  có phải là diễn viên an toàn hay không.
- split_blocks (bool , defaults to False) --
  Nếu Đúng, hãy tạo một "khối" nội bộ cho mỗi cột khi
  tạo pandas.DataFrame từ RecordBatch hoặc Table . Trong khi điều này
  có thể tạm thời giảm bộ nhớ lưu ý rằng các hoạt động khác nhau của gấu trúc
  có thể kích hoạt "hợp nhất" có thể sử dụng bộ nhớ bong bóng.
- self_destruct (bool , defaults to False) --
  THỬ NGHIỆM: Nếu Đúng, hãy cố gắng phân bổ lại Mũi tên ban đầu
  bộ nhớ trong khi chuyển đổi đối tượng Arrow thành gấu trúc. Nếu bạn sử dụng
  đối tượng sau khi gọi to_pandas bằng tùy chọn này, nó sẽ làm hỏng
  chương trình.
- types_mapper (function , defaults to None) --
  Một hàm ánh xạ Kiểu dữ liệu pyarrow tới ExtensionDtype của gấu trúc.
  Điều này có thể được sử dụng để ghi đè loại gấu trúc mặc định để chuyển đổi
  các loại pyarrow tích hợp hoặc không có pandas_metadata trong
  Lược đồ bảng. Hàm nhận được một Kiểu dữ liệu pyarrow và
  dự kiến sẽ trả về gấu trúc ExtensionDtype hoặc Không có nếu
  chuyển đổi mặc định nên được sử dụng cho loại đó. Nếu bạn có
  ánh xạ từ điển, bạn có thể chuyển dict.get dưới dạng function.0 pandas.Series hoặc
  pandas.DataFrame pandas.Series hoặc pandas.DataFrame tùy thuộc vào loại đối tượng

Chuyển đổi sang mảng NumPy hoặc DataFrame tương thích với gấu trúc, nếu thích hợp.

to_stringdatasets.table.ConcatenationTable.to_stringhttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L305[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]

fielddatasets.table.ConcatenationTable.fieldhttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L324[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- i

(Union[int, str]) --
Chỉ mục hoặc tên của trường cần truy xuất.0 pyarrow.Field

Chọn trường lược đồ theo tên cột hoặc chỉ mục số của nó.

cộtdatasets.table.ConcatenationTable.columnhttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L337[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- i (Union[int, str]) --
Chỉ mục hoặc tên của cột cần lấy.0 pyarrow.ChunkedArray

Chọn một cột theo tên cột hoặc chỉ mục bằng số.

itercolumnsdatasets.table.ConcatenationTable.itercolumnshttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L350[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}] pyarrow.ChunkedArray

Trình lặp trên tất cả các cột theo thứ tự số của chúng.

Schemadatasets.table.ConcatenationTable.schemahttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L359[] pyarrow.Schema

Lược đồ của bảng và các cột của nó.

cộtdatasets.table.ConcatenationTable.columnshttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L369[] List[pa.ChunkedArray]

Danh sách tất cả các cột theo thứ tự số.

num_columnsdatasets.table.ConcatenationTable.num_columnshttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L379[]int

Số cột trong bảng này.

num_rowsdatasets.table.ConcatenationTable.num_rowshttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L389[]int

Số hàng trong bảng này.

Do định nghĩa của một bảng, tất cả các cột có cùng số lượng
hàng.

Shapedatasets.table.ConcatenationTable.shapehttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L402[] (int, int) Number of rows and number of columns.

Dimensions of the table: (#rows, #columns).

nbytesdatasets.table.ConcatenationTable.nbyteshttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L412[]

Tổng số byte được sử dụng bởi các phần tử của bảng.

cột_namesdatasets.table.ConcatenationTable.column_nameshttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L419[]

Tên các cột của bảng.

slicedatasets.table.ConcatenationTable.slicehttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L1482[{"name": "offset", "val": " = 0"}, {"name": "length", "val": " = None"}]- offset (int , defaults to 0) --
Offset từ đầu bảng đến lát cắt.

- length (int , defaults to None) --
  Length of slice (default is until end of table starting from
  offset).0 datasets.table.Table

Tính toán phần không sao chép của Bảng này.

filterdatasets.table.ConcatenationTable.filterhttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L1513[{"name": "mask", "val": ""}, {"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]

Chọn các bản ghi từ một Bảng. Xem pyarrow.compute.filter để biết cách sử dụng đầy đủ.

Flattendatasets.table.ConcatenationTable.flattenhttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L1524[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- memory_pool (MemoryPool , defaults to None) --
Để phân bổ bộ nhớ, nếu được yêu cầu, nếu không, hãy sử dụng bộ dữ liệu pool.0 mặc định.table.Table

Làm phẳng cái bàn này. Mỗi cột có kiểu cấu trúc được làm phẳng
thành một cột cho mỗi trường cấu trúc. Các cột khác không thay đổi.

Combine_chunksdatasets.table.ConcatenationTable.combine_chunkshttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L1542[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- memory_pool (MemoryPool , defaults to None) --
Để phân bổ bộ nhớ, nếu được yêu cầu, nếu không, hãy sử dụng bộ dữ liệu pool.0 mặc định.table.Table

Tạo một bảng mới bằng cách kết hợp các khối mà bảng này có.

Tất cả các phần cơ bản trong ChunkedArray của mỗi cột đều
nối thành 0 hoặc một đoạn.

castdatasets.table.ConcatenationTable.casthttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L1562[{"name": "target_schema", "val": ""}, {"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- target_schema (Schema) --
Lược đồ để truyền tới, tên và thứ tự của các trường phải khớp nhau.

- safe (bool , defaults to True) --
  Kiểm tra tình trạng tràn hoặc các bộ dữ liệu chuyển đổi không an toàn khác.table.Table

Truyền các giá trị bảng sang một lược đồ khác.

thay_schema_metadatadatasets.table.ConcatenationTable.replace_schema_metadatahttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L1593[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- metadata (dict , defaults to None)
--0 bộ dữ liệu.table.Table nông_copy

THỬ NGHIỆM: Tạo bản sao nông của bảng bằng cách thay thế lược đồ
key-value metadata with the indicated new metadata (which may be None ,
which deletes any existing metadata).

add_columndatasets.table.ConcatenationTable.add_columnhttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L1611[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- i (int) --
Chỉ mục để đặt cột tại.

- field_ (Union[str, pyarrow.Field]) --
  Nếu một chuỗi được truyền thì loại được suy ra từ cột
  dữ liệu.
- column (Union[pyarrow.Array, List[pyarrow.Array]]) --

Cột data.0datadatas.table.Table Bảng mới đã thêm cột đã truyền.

Thêm cột vào Bảng tại vị trí.

Một bảng mới được trả về với cột được thêm vào, bảng gốc
đối tượng được giữ nguyên không thay đổi.

append_columndatasets.table.ConcatenationTable.append_columnhttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L1632[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- field_ (Union[str, pyarrow.Field]) --
Nếu một chuỗi được truyền thì loại được suy ra từ cột
dữ liệu.

- column (Union[pyarrow.Array, List[pyarrow.Array]]) --
  Cột data.0datadatas.table.Table Bảng mới đã thêm cột đã truyền.

Nối cột vào cuối cột.

Remove_columndatasets.table.ConcatenationTable.remove_columnhttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L1649[{"name": "i", "val": ""}, {"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- i (int) --
Chỉ mục của cột cần loại bỏ.0datadatas.table.Table Bảng mới không có cột.

Tạo Bảng mới với cột được chỉ định đã bị xóa.

set_columndatasets.table.ConcatenationTable.set_columnhttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L1673[{"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- i (int) --
Chỉ mục để đặt cột tại.

- field_ (Union[str, pyarrow.Field]) --
  Nếu một chuỗi được truyền thì loại được suy ra từ cột
  dữ liệu.
- column (Union[pyarrow.Array, List[pyarrow.Array]]) --
  Cột data.0datadatas.table.Table Bảng mới với tập hợp cột đã truyền.

Thay thế cột trong Bảng tại vị trí.

đổi tên_columnsdatasets.table.ConcatenationTable.rename_columnshttps://github.com/

huggingface/datasets/blob/4.2.0/src/datasets/table.py#L1692[{"name": "names", "val": ""}, {"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]

Tạo bảng mới với các cột được đổi tên thành tên được cung cấp.

selectdatasets.table.ConcatenationTable.selecthttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L1726[{"name": "columns", "val": ""}, {"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- columns (Union[List[str], List[int]]) --
Tên cột hoặc chỉ số nguyên cho bảng select.0datasets.table.TableNew với
các cột được chỉ định và siêu dữ liệu được bảo tồn.

Chọn các cột của bảng.

Trả về một bảng mới với các cột được chỉ định và siêu dữ liệu được giữ nguyên.

dropdatasets.table.ConcatenationTable.drophttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L1705[{"name": "columns", "val": ""}, {"name": "*args", "val": ""}, {"name": "**kwargs", "val": ""}]- columns (List[str]) --
Danh sách tên trường tham chiếu các cột hiện có.0 tập dữ liệu.table.Bảng Bảng mới không có cột.- KeyError -- : nếu bất kỳ tên cột nào được truyền không tồn tại. Lỗi phím

Bỏ một hoặc nhiều cột và trả về một bảng mới.

from_blocksdatasets.table.ConcatenationTable.from_blockshttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L1378[{"name": "blocks", "val": ": ~TableBlockContainer"}]

from_tablesdatasets.table.ConcatenationTable.from_tableshttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L1392[{"name": "tables", "val": ": list"}, {"name": "axis", "val": ": int = 0"}]- tables (list of Table or list of pyarrow.Table) --
Danh sách các bảng.

- axis ({0, 1} , defaults to 0 , meaning over rows) --
  Axis to concatenate over, where 0 means over rows (vertically) and 1 means over
  cột
  (horizontally).
  0
  Tạo ConcatenationTable từ danh sách các bảng.

Utilsdatasets.table.concat_tables

bộ dữ liệu.table.concat_tablesdatasets.table.concat_tableshttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L1746[{"name": "tables", "val": ": list"}, {"name": "axis", "val": ": int = 0"}]- tables (list of Table) --
Danh sách các bảng cần nối

- axis ({0, 1} , defaults to 0 , meaning over rows) --
  Axis to concatenate over, where 0 means over rows (vertically) and 1 means over cột
  (horizontally).
  0 datasets.table.Table If the number of input tables is > 1, then the returned table is a bộ dữ liệu.table.ConcatenationTable .
  Ngược lại, nếu chỉ có một bảng, nó sẽ được trả về nguyên trạng.

Nối các bảng.

bộ dữ liệu.table.list_table_cache_filesdatasets.table.list_table_cache_fileshttps://github.com/huggingface/datasets/blob/4.2.0/src/datasets/table.py#L1769[{"name": "table", "val": ": Table"}] List[str] A list of paths to the cache files loaded by the table.

Lấy các tập tin bộ đệm được tải bởi bảng.
Tệp bộ đệm được sử dụng khi các phần của bảng đến từ đĩa thông qua ánh xạ bộ nhớ.