1. Библиотеки / данные

Библиотека pandas позволяет выполнять слияние объектов с помощью операций, аналогичных операциям соединения для баз данных. Слияние полезно, поскольку оно позволяет нам связывать данные в разных объектах DаtaFrame с помощью значений, присутствующих в обоих наборах.

2. Слияние объектов DataFrame

Операция слияния или соединения комбинирует наборы данных, соединяя строки по одному или нескольким ключам. Эта операция является одной из основных в базах данных. Функция merge() в библиотеке pandas - портал ко всем алгоритмам такого рода.

По умолчанию функция merge() производит внутреннее соединение (INNER JOIN): в результирующий объект попадают только те ключи, которые присутствуют в обоих объектах.

Если имена столбцов в объекте различается, то можно задать их порознь с помощью параметров left\_on и right\_оn.

У функции merge() параметр suffixes позволяет задать строки, которые будут записываться в конец одинаковых имен в левом и в правом объекте DataFrame.

Для слияния по нескольким ключам следует передать список имен столбцов. Так же используем параметры left\_on и right\_оn.

Как уже упоминалось, по умолчанию функция merge() производит внутреннее соединение. Альтернативы inner – это left, right и outer. В случае внешнего соединения - outer - берется объединение ключей, то есть получается то же самое, что при совместном применении левого и правого соединения (FULL OUTER JOIN).

Иногда ключ или ключи слияния находятся в индексе объекта DataFrame. В таком случае можно задать параметр left\_index = True или right\_index = True или и то, и другое.

3. Конкатенация объектов DataFrame

Еще одну операцию комбинирования данных разные авторы называют по-разному: конкатенация, связывание или укладка.

В контексте объектов Series и DataFrame наличие помеченных осей позволяет обобщить конкатенацию массивов. В частности, нужно решить следующие вопросы: если объекты по-разному проиндексированы, то как поступать с коллекциями меток: объединять или пересекать; нужно ли иметь возможность идентифицировать группы в результирующем объекте, имеет ли вообще какое-то значение ось конкатенации. Функция concat() дает согласованные ответы на эти вопросы.

Если два объекта DataFrame передать функции concat() списком, то она склеит их данные построчно. Чтобы сохранить вывод, добавим параметр sort = True. Данный параметр отвечает за сортировку имен столбцов. Важно отметить, что в этом примере происходит выравнивание по меткам столбцов. В итоговый объект DataFrame входят все уникальные столбцы из двух исходных объектов DataFrame.

Но можно образовать и пересечение, передав параметр join = ‘inner’. В результирующем объекте DataFrame появляются столбцы, которые присутствуют в обоих объектах DataFrame.

Eсли передать параметр axis = 1, то конкатенация произойдет по столбцам, выравнивание - по меткам индекса строк.

4. "Long" и "Wide" форматы

Данные часто хранятся в состыкованном формате, который еще называется Long-форматом. Он используется в базах данных, csv-файлах и таблицах Excel.

В качестве примера возьмем поток данных, регистрируемых акселерометром. Производится измерение ускорения по трем осям в разные промежутки времени. Промежутки времени записываются в столбец interval, метки оси - в столбец axis, ускорения - в столбец reading. Проблема такого представления данных заключается в том, что довольно трудно выделить показания, относящиеся к конкретной оси. Это можно сделать помощью метода unstack(), предварительно поместив interval и axis в индекс строк. Метод unstack() поворачивает уровень меток строк, превращая его в индекс столбцов.

Еще один способ преобразовать данные из Long-формата в Wide-формат - это использование метода pivot().

Метод stack() выполняет обратное действие unstack(), то есть поворачивает уровень индекса столбцов, превращая его в уровень индекса строк.

Получив такое представление данных, мы можем получить все показания по всем осям и по всем пользователям в интервале 1 с помощью метода xs(). Метод unstack() помещает самый внутренний уровень индекса строк в новый уровень индекса столбцов. В результате получаем столбцы с типом индексации «мультииндекс».

Чтобы применить unstack() для другого уровня, используем параметр level. Первый уровень, то есть interval, будет помещен в ось столбцов.

Есть еще пара вещей, о которых стоит упомянуть, глядя на полученный результат. Во-первых, методы stack() и unstack() всегда помещают уровни в самые внутренние уровни другого индекса. Во-вторых, при перемещении данных, которое осуществляют методы stack() и unstack() не происходит потеря информации. При выполнении этих операций просто меняется способ организации и просмотра данных.