pandas

Jan Popko

Python Advanced

Business Trends Academy (BTA) GmbH

bta
BUSINESS TRENDS ACADEMY

Nestorstraße 36
D-10709 Berlin

Tel.: +49 (0) 30 894 087 57
Fax: +49 (0) 30 895 429 94

Geschäftsführer:
Gabriele Fleischmann-Hahn
Maxi-Marlen Fleischmann
Hauptsitz des Unternehmens:
Nestorstraße 36, D-10709 Berlin
HRB 115251 B / Amtsgericht Berlin Charlottenburg
Steuer-Nr. 27/248/31179

pandas

result = pd.concat([df1, df4], axis=1, sort=False)

df1

	A	B	C	D
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3

df4

	B	D	F
2	B2	D2	F2
3	B3	D3	F3
6	B6	D6	F6
7	B7	D7	F7

Result

	A	B	C	D	B	D	F
0	A0	B0	C0	D0	NaN	NaN	NaN
1	A1	B1	C1	D1	NaN	NaN	NaN
2	A2	B2	C2	D2	B2	D2	F2
3	A3	B3	C3	D3	B3	D3	F3
6	NaN	NaN	NaN	NaN	B6	D6	F6
7	NaN	NaN	NaN	NaN	B7	D7	F7

pandas

Konkatination

pd.concat() - Verbindet zwei pandas Datentypen (ähnlich der numpy.concatenate()

Parameter:
- axis - 0/1 oder 'index'/'columns', sollen Spalten oder Zeilen angeähngt werden
- ignore_index - True/False, die Indices der Zeilen/Spalten werden ignoriert
- verify_integrity - True/False, testen ob die angehängte Achse Duplikate enthält
  - kann sehr rechenaufwändig sein
- join - 'inner'/'outer' - wie werden Indices der anderen Achsen gehändelt
- sort - True/False/None - Explizit die nicht-Konkatenationsachse Sortieren

Für MultiIndex:
- name - Name der Levels für hierarchische Indices
- keys - erstelle hierarchische Indices (als Tupel übergeben)
- levels - spezifiere die Levels um MultiIndex zu erstellen (sonst über keys)

Pandas - merge

pd.merge() - verbindet zwei pandas DataFrames an einer bestimmten Spalte

Parameter:
- on - Spaltenname, gibt an, über welche Spalten die Verbindung erstellt werden soll (die Namen müssen für beide DataFrames gleich sein)
- left_on - gibt an, welche Spalte für die Verbindung genutzt wird (erstes DataFrame)
- left_on - gibt an, welche Spalte für die Verbindung genutzt wird (zweites DataFrame)
- left_index - True/False, gibt an, ob der Index als Verbindungsschlüssel genutzt werden soll (erstes DataFrame)
- right_index - True/False, gibt an, ob der Index als Verbindungsschlüssel genutzt werden soll (zweites DataFrame)
    - es ist auch möglich eine Spalte und eine Achse als Verbindungsschlüssel zu nutzen

- suffixes - Tuple (str, str), wenn zwei Spalten gleich heißen, dann wird der String im Tupel an die jeweilige Stelle angehängt

Pandas - merge

Parameter 'validate':

- 'one_to_one'/'1:1' – testet ob die Verbindungskeys in beiden DataFrames einzigartig sind

- 'one_to_many'/'1:m' – tested ob die Verbindungskeys im ersten DataFrame einzigartig sind

- 'many_to_many'/'m:m' – tested nichts (Verbindungskeyes können in beiden DataFrames mehr als einmal vorkommen)

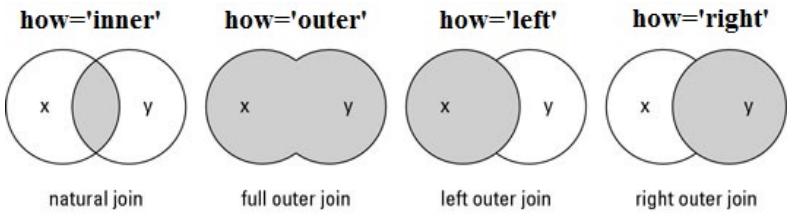
Die Art der Verbindungen wird automatisch erstellt.
'validate' testet nur ob eine bestimmte Verbindung überhaupt möglich ist.

Pandas - join

DataFrame.join() - verbindet DataFrames ähnlich wie SQL-Datenbanken verbunden werden über ihren Index

Parameter 'how':

- 'inner' – nur die Indices, die in beiden DataFrames vorkommen werden übernommen
- 'outer' – alle Indices werden übernommen (Werte ggf. mit NaN ersettz.)
- 'left' – nur die Indices aus dem ersten Dataframe werden übernommen
- 'right' – nur die Indices aus dem zweiten Dataframe werden übernommen



pandas

Umgang mit fehlenden Daten

Testen auf fehlende Daten:
`isnull()` - erstellt boolsche Maske mit True für fehlende Daten
`notnull()` - erstellt boolsche Maske mit False für fehlende Daten

Entfernen von fehlenden Daten:
`dropna()` - entfernt alle Zeilen und Spalten mit NaN Werten
Parameter: axis = 0/1 → suche nur über Zeilen oder Spalten
how = 'any' → Entfernt wenn ein Eintrag als NaN Wert gesetzt ist
how = 'all' → Entfernt wenn alle Einträge als NaN Wert gesetzt ist
(how hängt von axis ab)

`fillna()` - füllt alle NaN Werte mit einem bestimmten Wert auf
Parameter: axis = 0/1 → Befüllung der Werte über eine bestimmte Achse
method = 'ffill' → Befüllung mit dem vorherigen Wert auf der Achse
method = 'bfill' → Befüllung mit dem nachfolgendem Wert auf der Achse

pandas

Umgang mit fehlenden Daten

Durch setzen einer "Sentinel Value"

Typeclass Conversion:

floating	No change	np.nan
object	No change	None or np.nan
integer	Cast to float64	np.nan
boolean	Cast to object	None or np.nan

Business Trends Academy (BTA) GmbH
Nestorstraße 36
D-10709 Berlin
Tel.: +49 (0) 30 894 087 57
Fax: +49 (0) 30 895 429 94
Geschäftsführer:
Gabriele Fleischmann-Hahn
Maxi-Marlen Fleischmann
Hauptsitz des Unternehmens:
Nestorstraße 36, D-10709 Berlin
HRB 115251 B / Amtsgericht Berlin Charlottenburg
Steuer-Nr. 27/248/31179