



Dealing with non-exact matches

- Common (manual) error: typos, upper/lower case → fuzzy string searching → **Fuzzy match** step
- Example:

stream of data with incorrect states
Data Grid step

Details of the fuzzy search

Step name	mylist
Meta	# my_state
Data	1 California
	2 Colorado
	3 Washington
	4 Massachusetts
	5 Alsaka
	6 Conneticut
	7 Road Island
	8 Hawai
	9 Ohio
	10 Kentucky
	11 Pennsylvania
	12 Louisiana

states_of_usa.txt	UNREK
1	State;Abbreviation
2	Alabama;AL
3	Alaska;AK
4	Arizona;AZ
5	Arkansas;AR

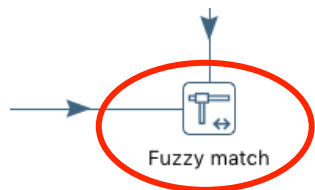
Algorithm	Levenshtein
Case sensitive	<input type="checkbox"/>
Get closer value	<input checked="" type="checkbox"/>
Minimal value	0
Maximal value	1
Values separator	,

Match field	match
Value field	distance



Dealing with non-exact matches

- Common (manual) error: typos, upper/lower case → fuzzy string searching → **Fuzzy match** step
- Example:



Details of the fuzzy search

Fuzzy string search

Step name: Fuzzy match

General Fields

Lookup stream (source)

Lookup step: states of usa
Lookup field: State

Main stream

Main stream field: mv.state

Settings

Algorithm: Levenshtein
Case sensitive:
Get closer value:
Minimal value: 0
Maximal value: 1
Values separator: ,

Help OK Cancel

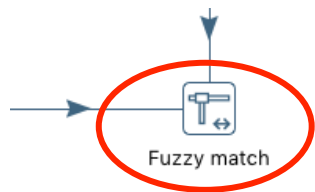
Several matching algorithms:

- Algorithms based on a **metric distance**, the comparison is based on how the terms are spelt (e.g. **Levenshtein**: This algorithm calculates the distance between two strings as the number of edit steps (character insertion or deletion, or replacements) needed to get from one string to another)
- Phonetic algorithms, the comparison is based on how the terms **sound** when read in English



Dealing with non-exact matches

- Common (manual) error: typos, upper/lower case → fuzzy string searching → **Fuzzy match** step
- Example:



Examine preview data

Rows of step: Fuzzy match (17 rows)

#	my_state	match	distance
1	California	<null>	<null>
2	Colorado	Colorado	1
3	Washington	Washington	0
4	Masachusetts	Massachusetts	1
5	Alsaka	<null>	<null>
6	Conneticut	Connecticut	1
7	Road Island	<null>	<null>
8	Hawai	Hawaii	1

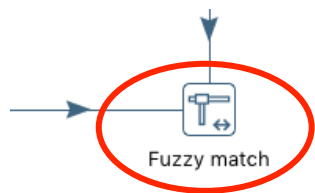
Close

- We used **Levenshtein algorithm**: This algorithm calculates the distance between two strings as the number of edit steps (character insertion or deletion, or replacements) needed to get from one string to another
- Try the **Damerau-Levenshtein** algorithm, which is similar to Levenshtein but adds the transposition operation



Deduplicating non-exact matches

- Example: **Hawaii**, **Hawaii**, and **Howaii** → We only want a single state: **Hawaii**



sort

Examine preview data

Rows of step: Fuzzy match (17 rows)

#	my_state	match	distance
15	Arizona	Arizona	1
16	California	California	0
2	Colorado	Colorado	1
6	Conneticut	Connecticut	1
8	Hawai	Hawaii	1
14	Hawaii	Hawaii	0
17	Howaii	Hawaii	1
10	Kentucky	Kentucky	1
12	Louisiana	Louisiana	0

Close



Deduplicating non-exact matches

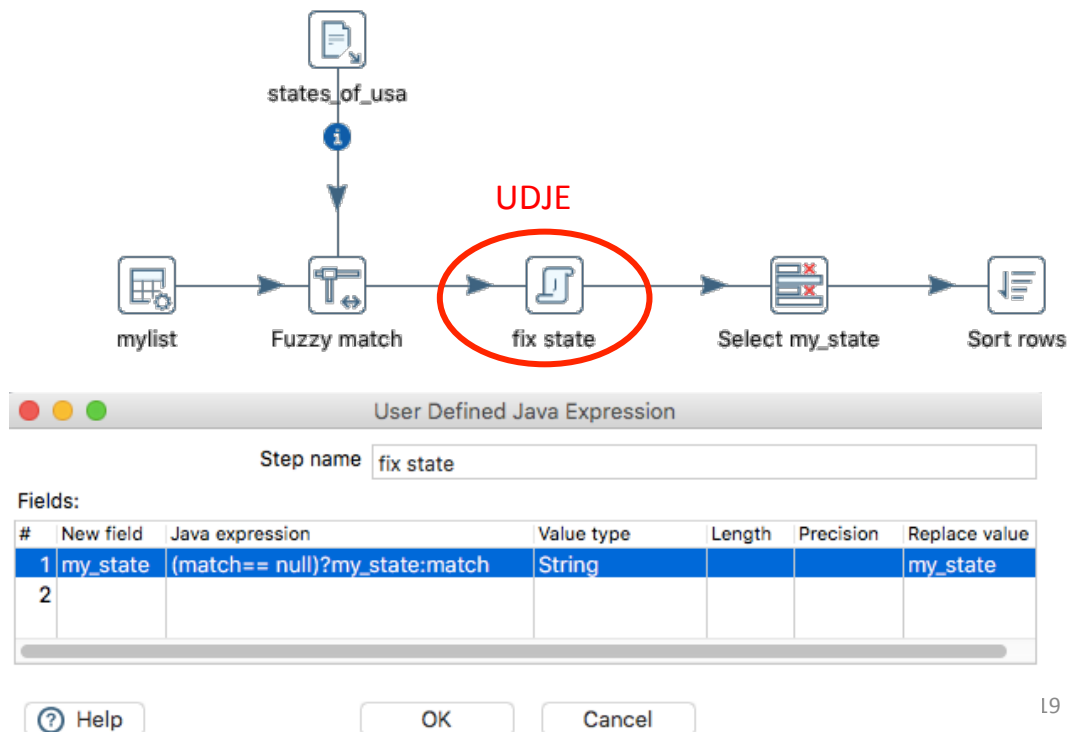
- Example: **Hawaii**, **Hawaii**, and **Howaii** → We only want a single state: **Hawaii**

Examine preview data

Rows of step: fix state (17 rows)

#	my_state	match	distance
5	Alaska	Alaska	1
13	Arizona	Arizona	1
1	California	California	1
16	California	California	0
2	Colorado	Colorado	1
6	Connecticut	Connecticut	1
8	Hawaii	Hawaii	1
14	Hawaii	Hawaii	0
17	Hawaii	Hawaii	1
10	Kentucky	Kentucky	1

Close





Deduplicating non-exact matches

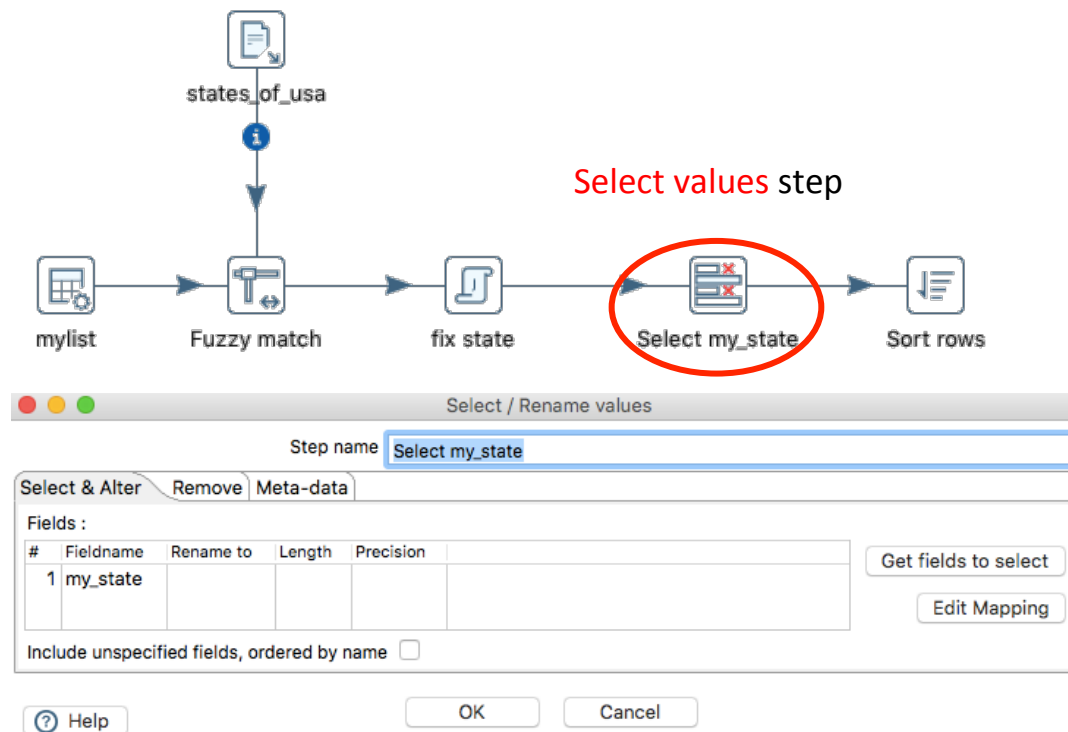
- Example: **Hawaii**, **Hawai**, and **Howai** → We only want a single state: **Hawaii**

Examine previ...

Rows of step: Select my_state

#	my_state
5	Alaska
13	Arizona
1	California
16	California
2	Colorado
6	Connecticut
8	Hawaii
14	Hawaii
17	Hawaii
10	Kentucky
12	Louisiana

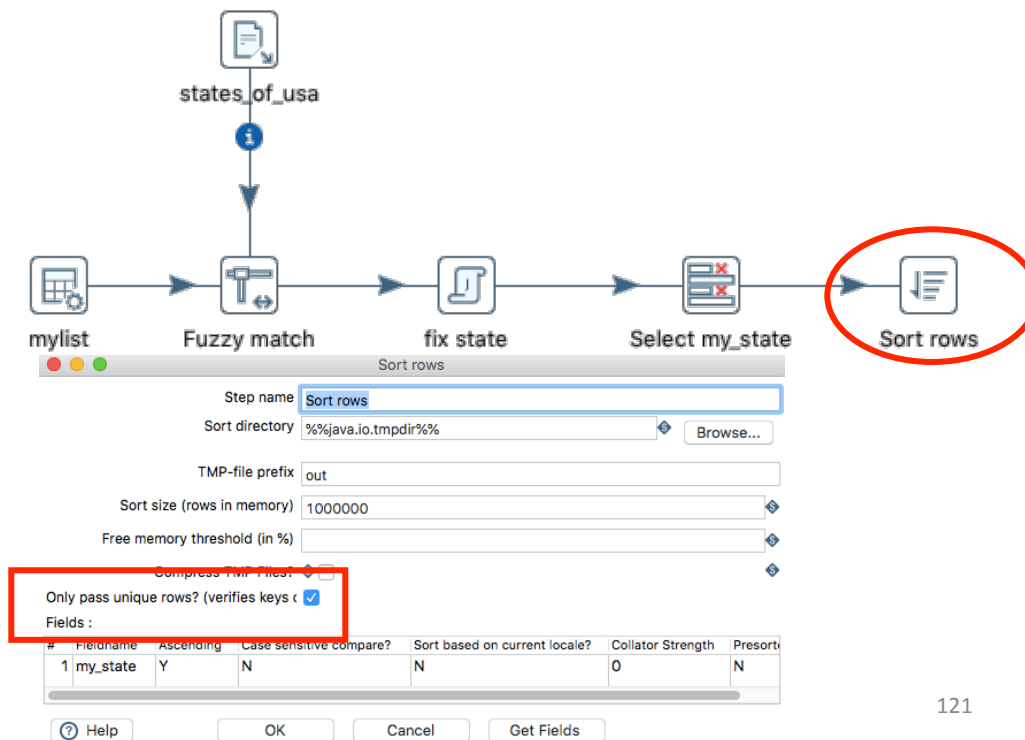
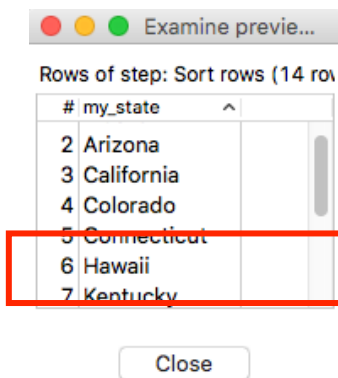
Close





Deduplicating non-exact matches

- Example: **Hawaii**, **Hawaii**, and **Howaii** → We only want a single state: **Hawaii**



Remove duplicates
We eliminated duplicated rows with the **Sort rows** step. If the data were sorted by state, we could have used a **Unique rows** step instead.