

EXAMPLE - Extractkv and Unnest Transforms

This example shows how you can unpack data nested in an Object into separate columns using the following transforms:

- `extractkv` - Removes key-value pairs from a source string. See *Extract Transform*.
- `unnest` - Unpacks nested data in separate rows and columns. See *Unnest Transform*.

Source:

You have the following information on used cars. The `VIN` column contains vehicle identifiers, and the `Properties` column contains key-value pairs describing characteristics of each vehicle. You want to unpack this data into separate columns.

VIN	Properties
XX3 JT4522	year=2004,make=Subaru,model=Impreza,color=green,mileage=125422,cost=3199
HT4 UJ9122	year=2006,make=VW,model=Passat,color=silver,mileage=102941,cost=4599
KC2 WZ9231	year=2009,make=GMC,model=Yukon,color=black,mileage=68213,cost=12899
LL8 UH4921	year=2011,make=BMW,model=328i,color=brown,mileage=57212,cost=16999

Transform:

Add the following transform, which identifies all of the key values in the column as beginning with alphabetical characters.

- The `valueafter` string identifies where the corresponding value begins after the key.
- The `delimiter` string indicates the end of each key-value pair.

```
extractkv col:Properties key: `[alpha]+` valueafter: `=` delimiter: `,`
```

Now that the Object of values has been created, you can use the `unnest` transform to unpack this mapped data. In the following, each key is specified, which results in separate columns headed by the named key:

```
unnest col:extractkv_Properties keys: 'year', 'make', 'model', 'color', 'mileage', 'cost'
```

Results:

When you drop the unnecessary `Properties` columns, the dataset now looks like the following:

VIN	year	make	model	color	mileage	cost
XX3 JT4522	2004	Subaru	Impreza	green	125422	3199
HT4 UJ9122	2006	VW	Passat	silver	102941	4599
KC2 WZ9231	2009	GMC	Yukon	black	68213	12899
LL8 UH4921	2011	BMW	328i	brown	57212	16999