IMEN891K / IDSC501 / AXGS-501 · DATA ENGINEERING · FALL 2026

4.

Pre-processing I: Cleaning & Integration

Sangwoo Mo · POSTECH

Industrial & Management Engineering

Where We Are

PART I · Collection and Preparation | **PART II** | **PART III**

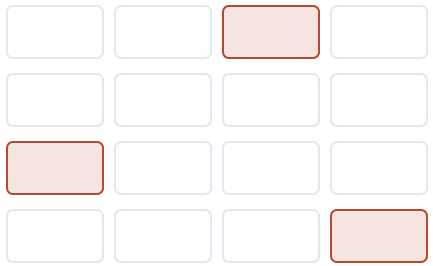


nothing downstream can recover what preprocessing throws away

Outline: Four Stages of Preprocessing

01

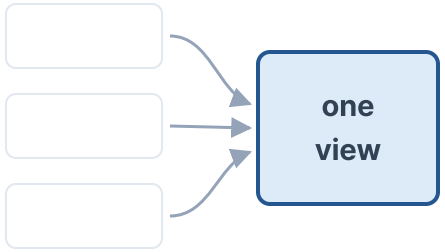
Cleaning



Wrong values in one table

02

Integration



Tables that disagree

03

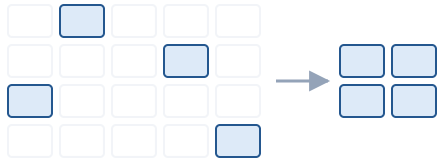
Transformation



Values a model can read

04

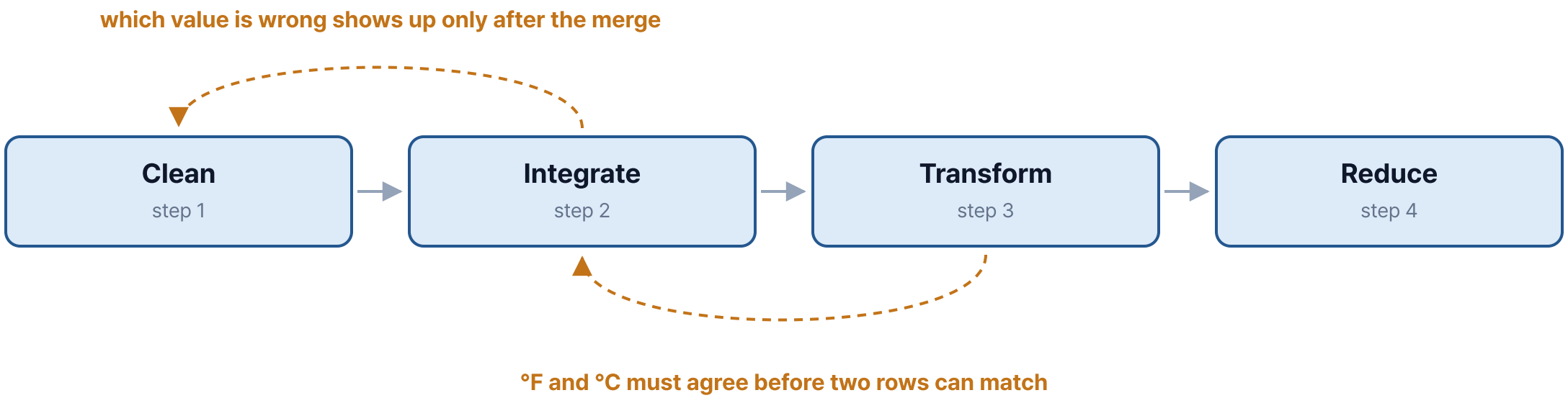
Reduction



Less data, same signal

Not a One-Way Pipeline

Two of today's sections are steps running out of order

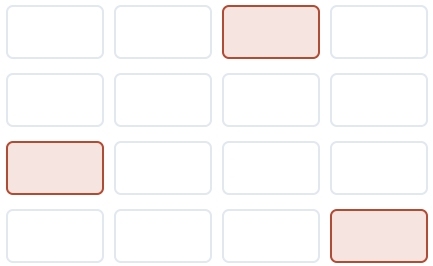


Both orange arrows run backward, and both are sections of part 2

Cleaning

01

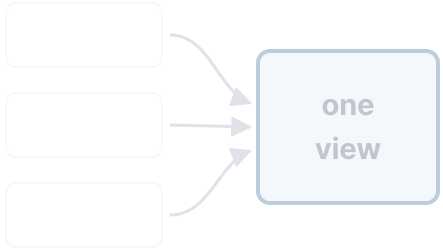
Cleaning



Wrong values in one table

02

Integration



Tables that disagree

03

Transformation



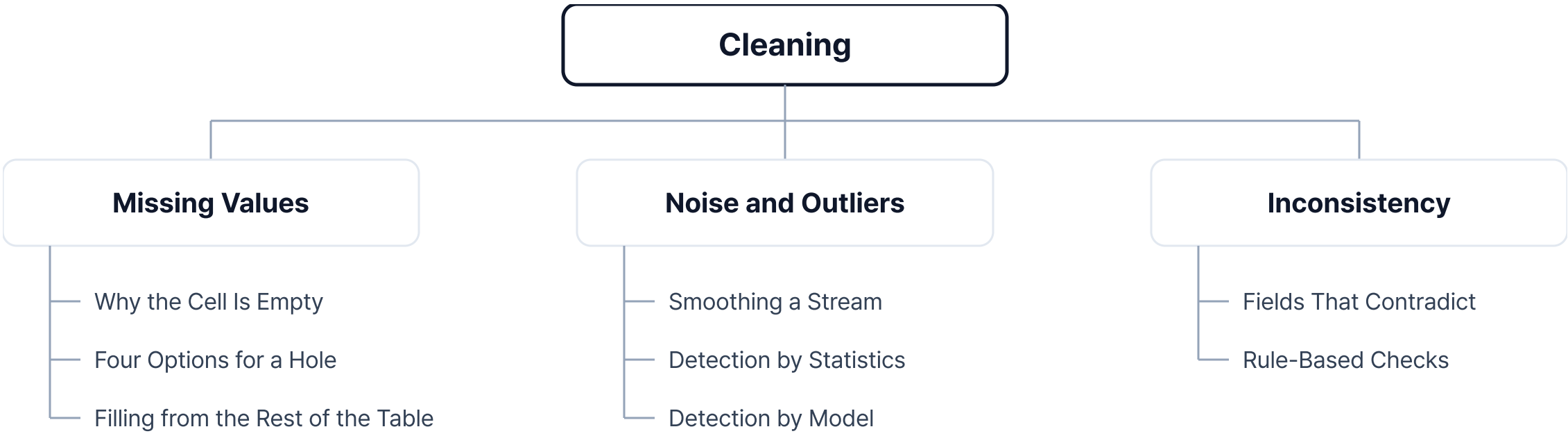
Values a model can read

04

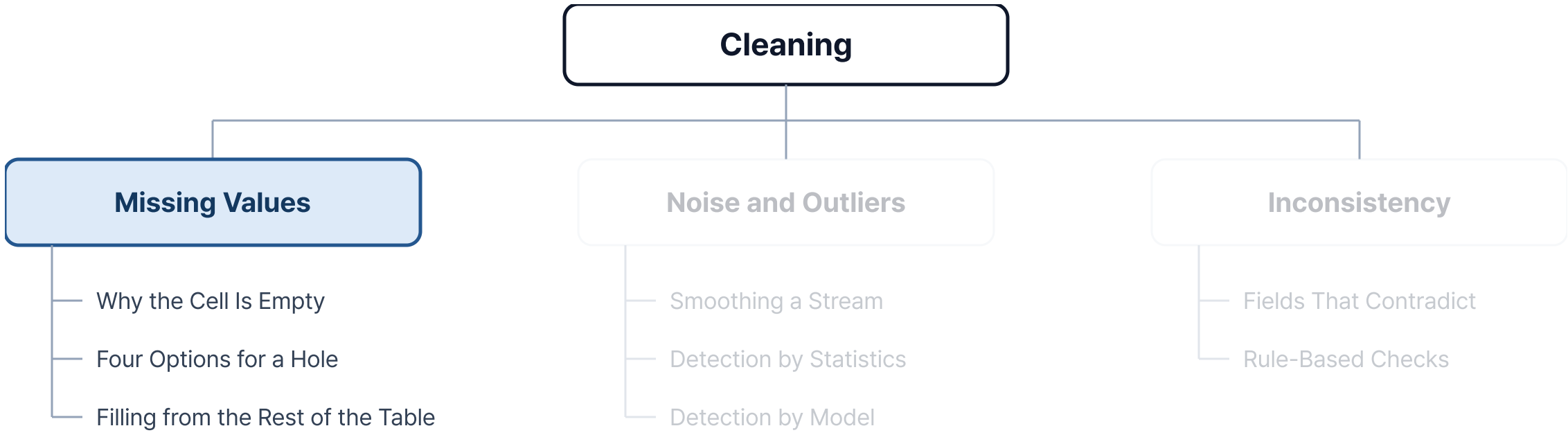
Reduction

Less data, same signal

Three Branches of Cleaning



Missing Values



Why the Cell Is Empty

ROW	STATE	SALARY	EXPERIENCE
1	NY	57,400	Mid
2	TX	**missing**	Entry
3	NJ	90,000	High
4	VT	36,900	Entry
5	TX	**missing**	Mid
6	CA	76,600	High
7	NY	85,000	High
8	CA	**missing**	Entry

Different missingness causes need different remedies

Caveat: Why is it missing?

Holes independent of the value

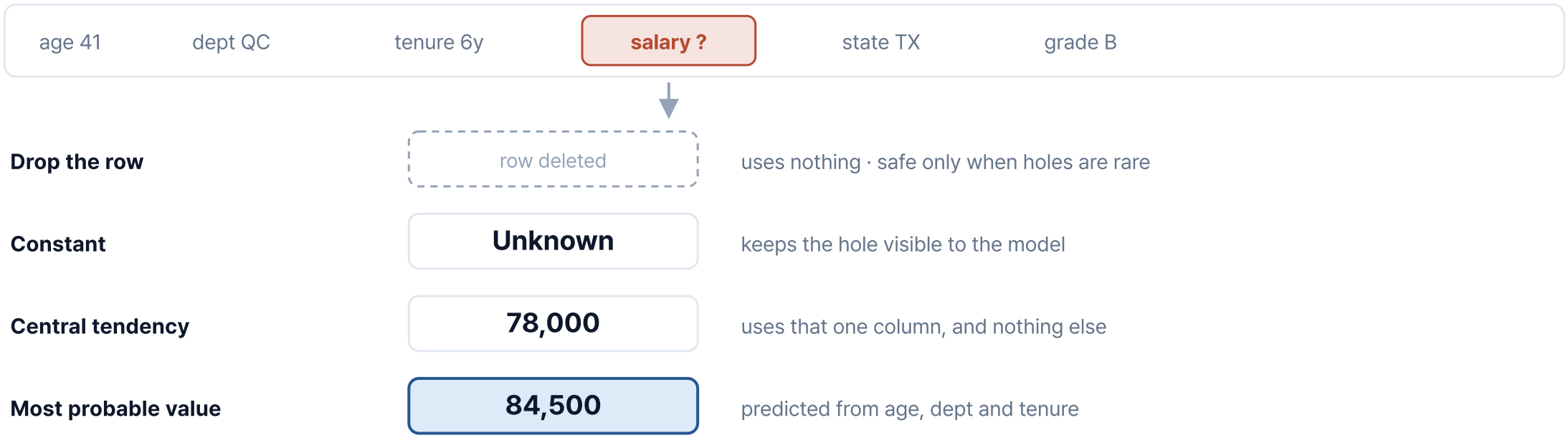
age	25	26	29	30	30	31	44	46	48	51	52	54
IQ	×	121	91	×	110	×	118	93	×	×	116	×

Holes concentrated in the younger half

age	25	26	29	30	30	31	44	46	48	51	52	54
IQ	×	×	×	×	×	×	118	93	116	141	97	104

Below, no young IQ was ever observed. **No imputer fixes this.** The question is not which method to use but why those rows are missing, and that answer lives in how the data was collected

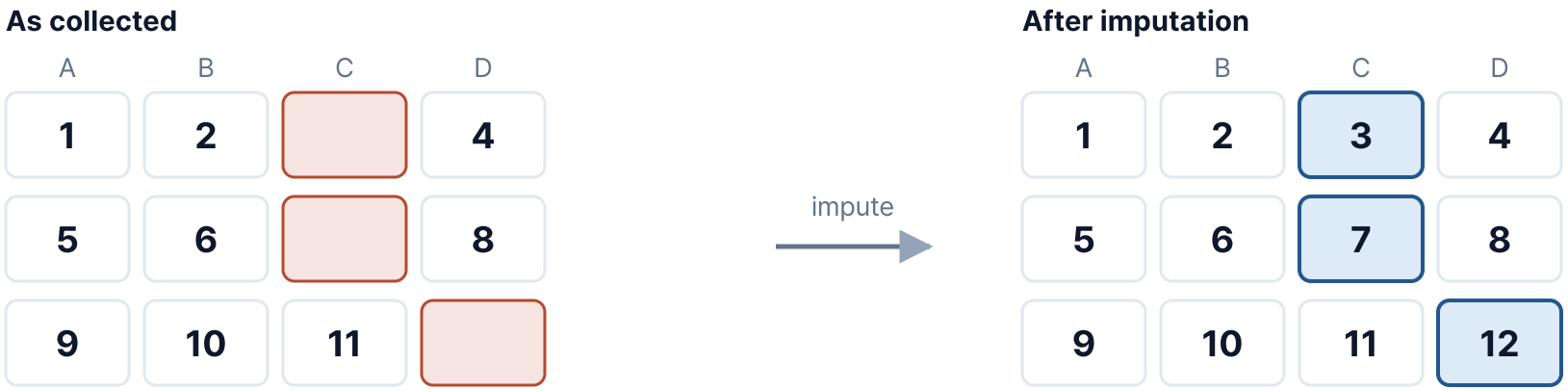
Four Options for a Hole



Top to bottom, **each one uses more of the rest of the record**

Imputation as Estimation

Filling a hole from the rest of the data

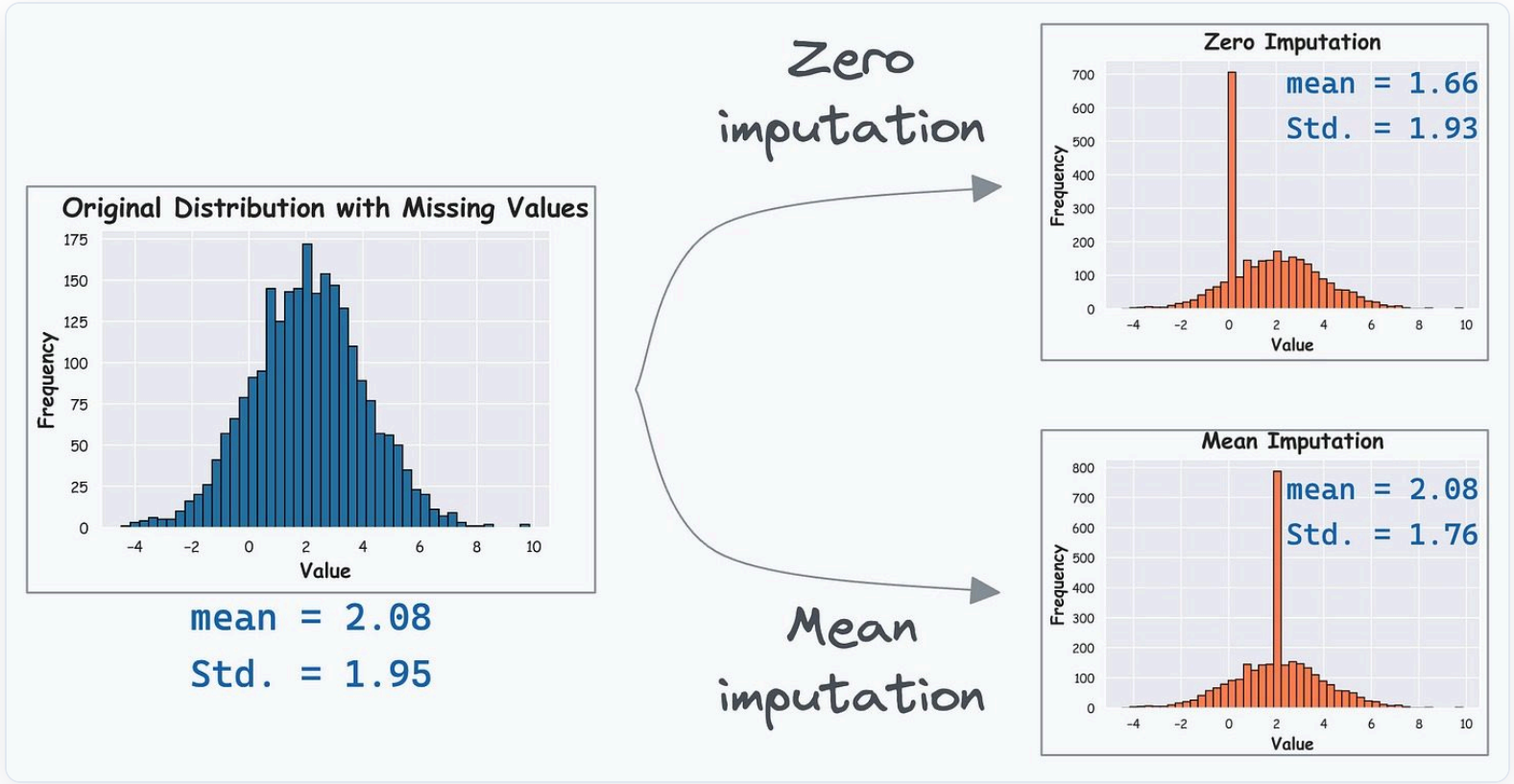


$\hat{x}_{ij} = f(\ X_{observed}\ )$ every filled cell is a prediction from the rest of the table

Every filled cell is an estimate, not an observation

Caveat: Mean Imputation

Every hole gets the same value, and the spread of the column collapses



Mean stays 2.08; standard deviation falls to 1.76 and a spike appears

Observed and Filled Values

Use other observed columns to find comparable rows

Observed measurements

part	helper	target
P1	10	20
P2	12	24
P3	11	NULL

Estimate for P3

Neighbors in helper: P1 and P2

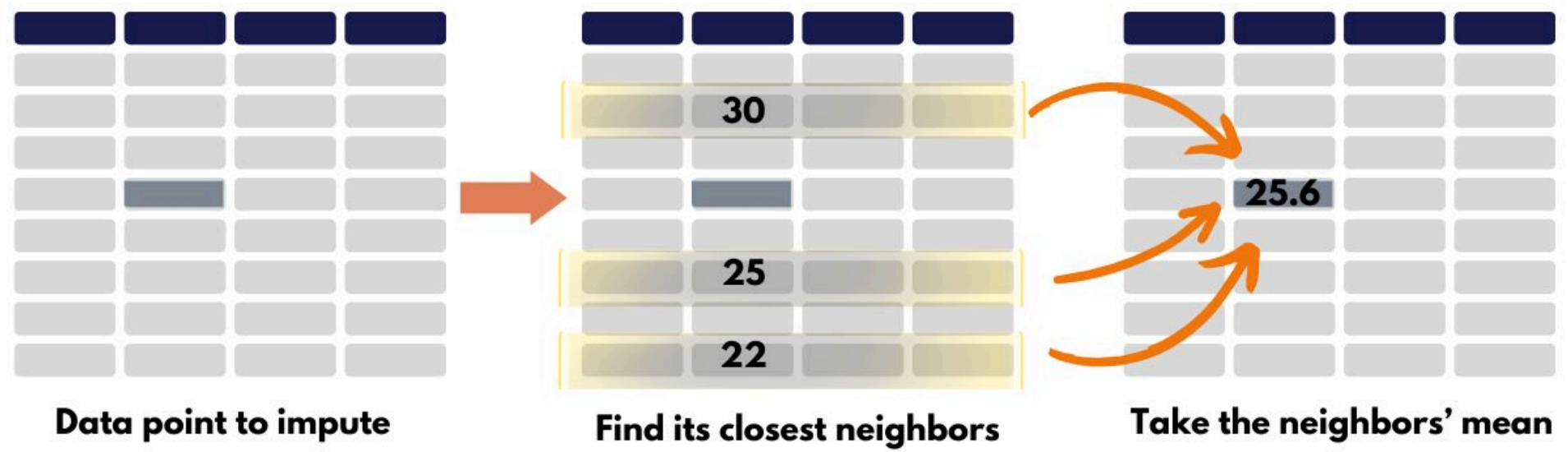
(20 + 24) / 2 = 22

Stored with provenance

part	target	estimated
P3	22	Yes

The filled number is an estimate; keep its origin distinguishable

k-Nearest Neighbor (kNN) Imputation



$\hat{x}_{ij} = (1/k) \sum_{r \in N_k(i)} x_{rj}$ closest in the other columns, or in an embedding of them

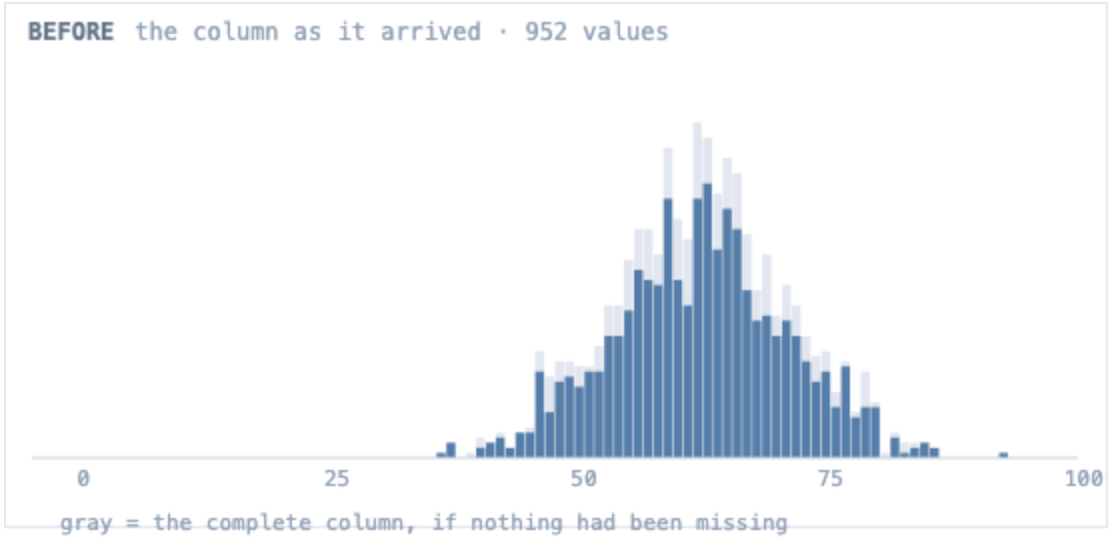
Average similar rows; including all observed rows gives the column mean

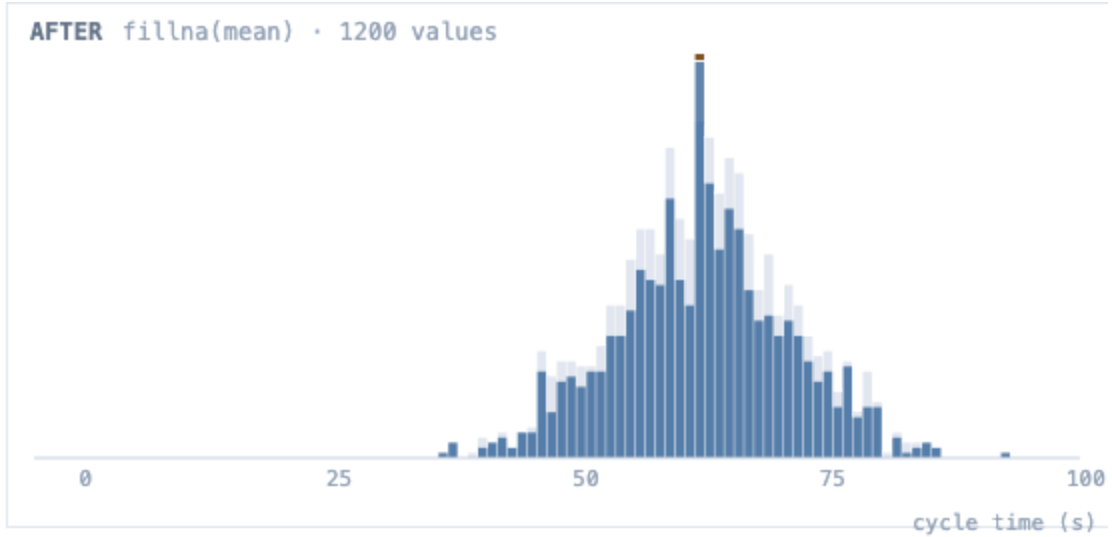
Demo: Imputation Methods

One column of cycle times, 1,200 rows: every fill changes the column

1. Compare mean and kNN fill. 2. What changes when the helper column contains little information?

Missing 20% Helper column ρ 0.75 Method fillna(mean)





true **μ 61.7 σ 8.9** · after **μ 61.7 σ 7.9** · rows **1200**

KNNImputer fills a hole from rows that are close in a second column. ρ is how much that column knows about this one. At ρ 0 it knows nothing.

spread −11% · mean −0.0

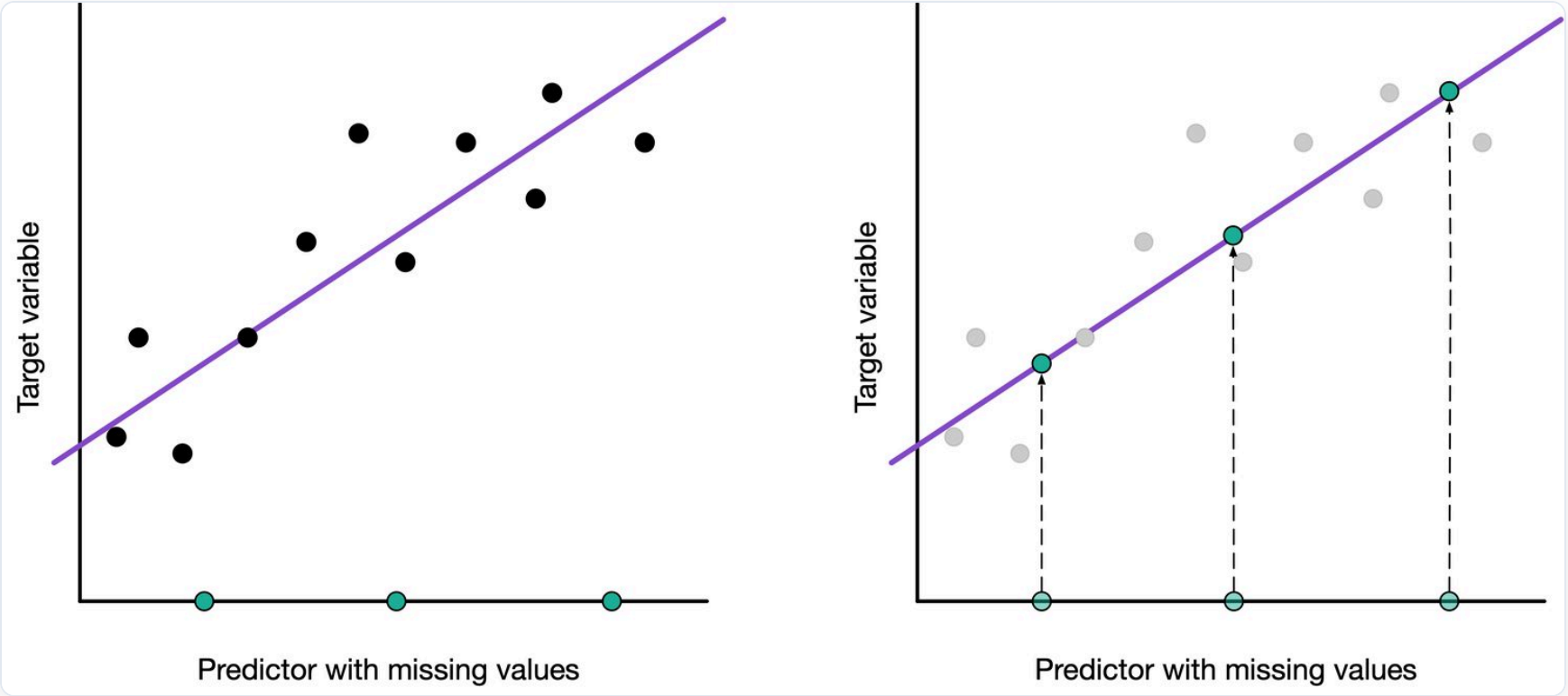
Answer: Imputation Choices

Compare distributions, not only the filled cells

Choice	What to inspect
`Mean fill`	`Mean can stay close; variance shrinks`
`kNN fill`	`Does the helper column contain information?`
`Missingness mask`	`Keep observed and estimated cells distinguishable`

A filled cell is an estimate, not a new observation

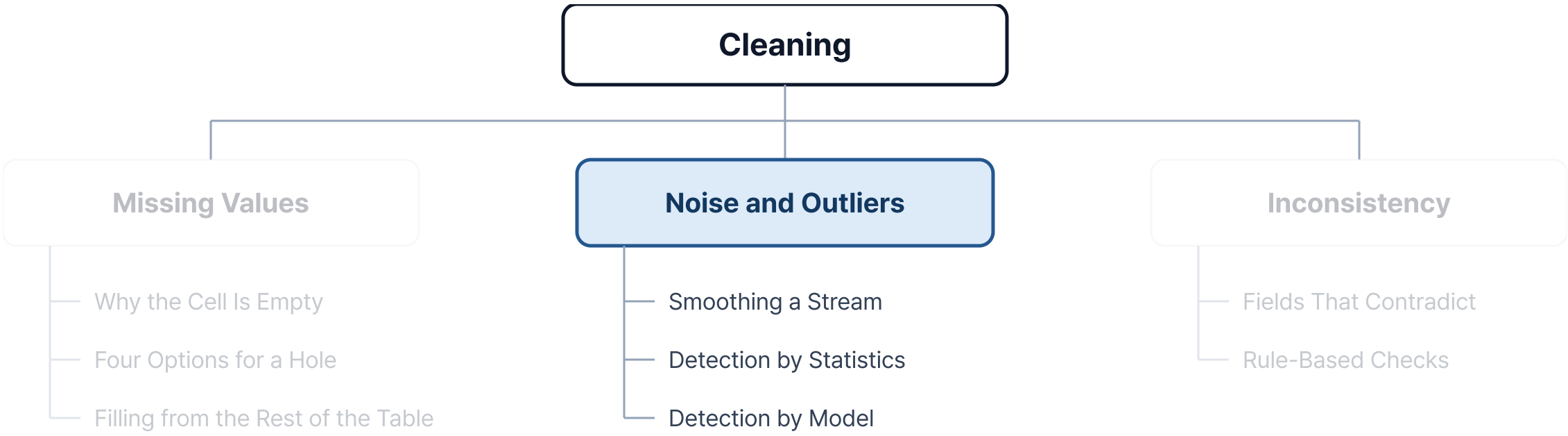
Regression Imputation



$$\hat{x}_{ij} = b_0 + \sum_{m \neq j} b_m x_{im}$$ no residual is added back, so the spread shrinks

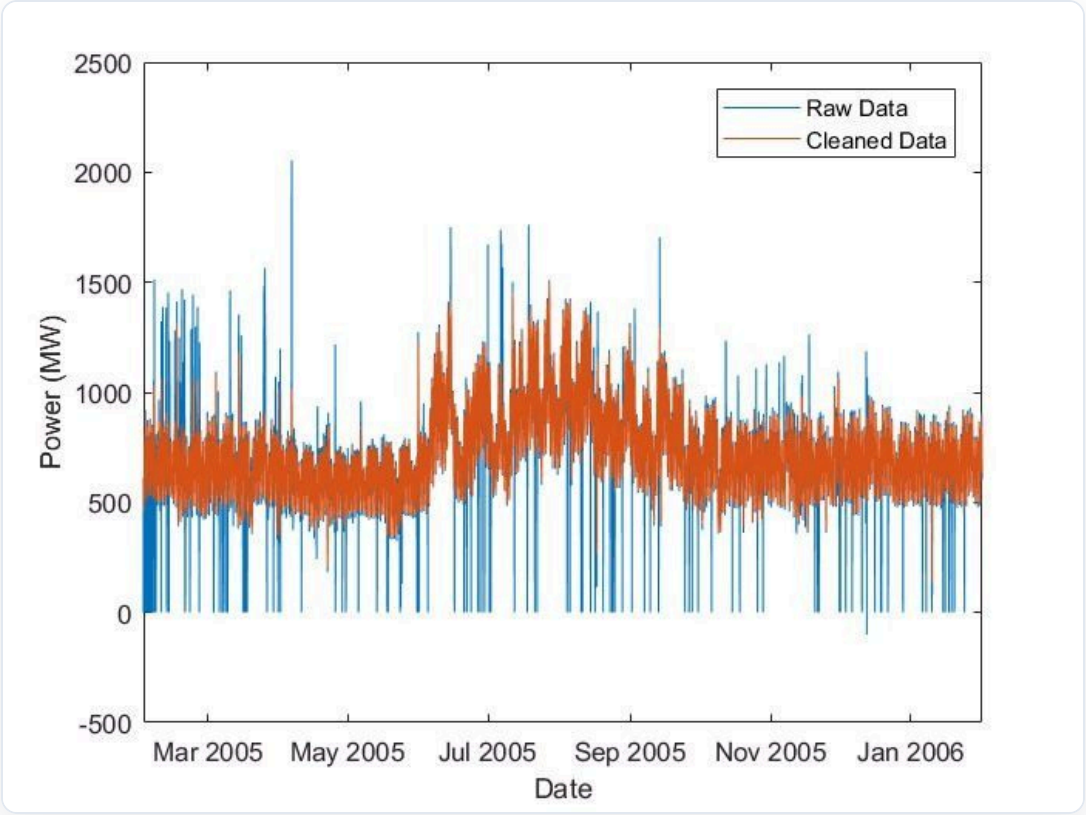
Deterministic predictions remove residual variation and can make correlation look stronger

Noise and Outliers



Raw Sensor Streams

Metered power demand, with two different kinds of error mixed in

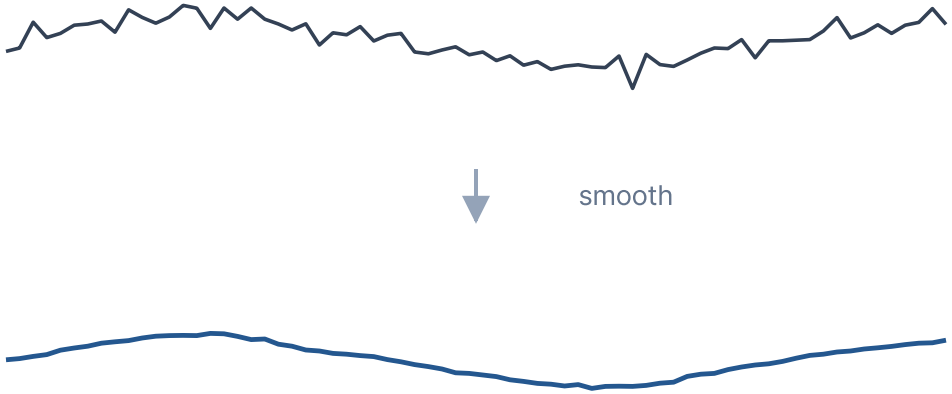


Blue spikes drop to zero and below. **That is the meter dropping out, not demand**

Noise vs Outliers

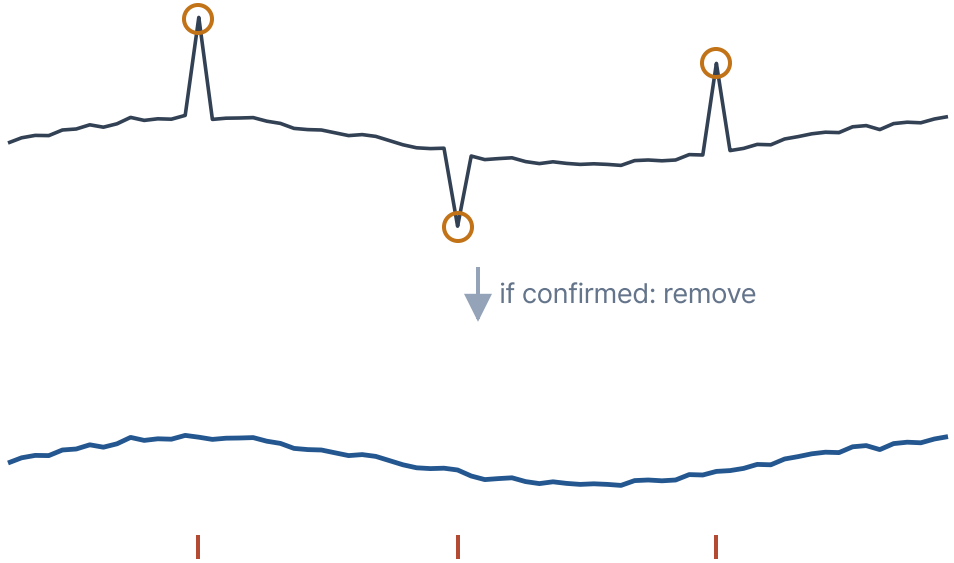
Smooth measurement noise; investigate outliers before changing them

Noise: every value is a little wrong



all 1,200 rows survive

Outliers: a few values sit far away

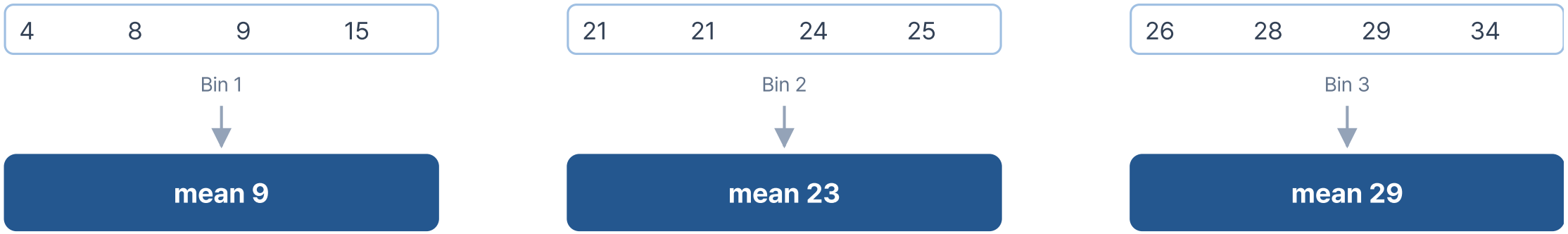


three rows are gone

Smoothing changes values; removal drops rows

Smoothing by Binning

Sorted values (price, $)



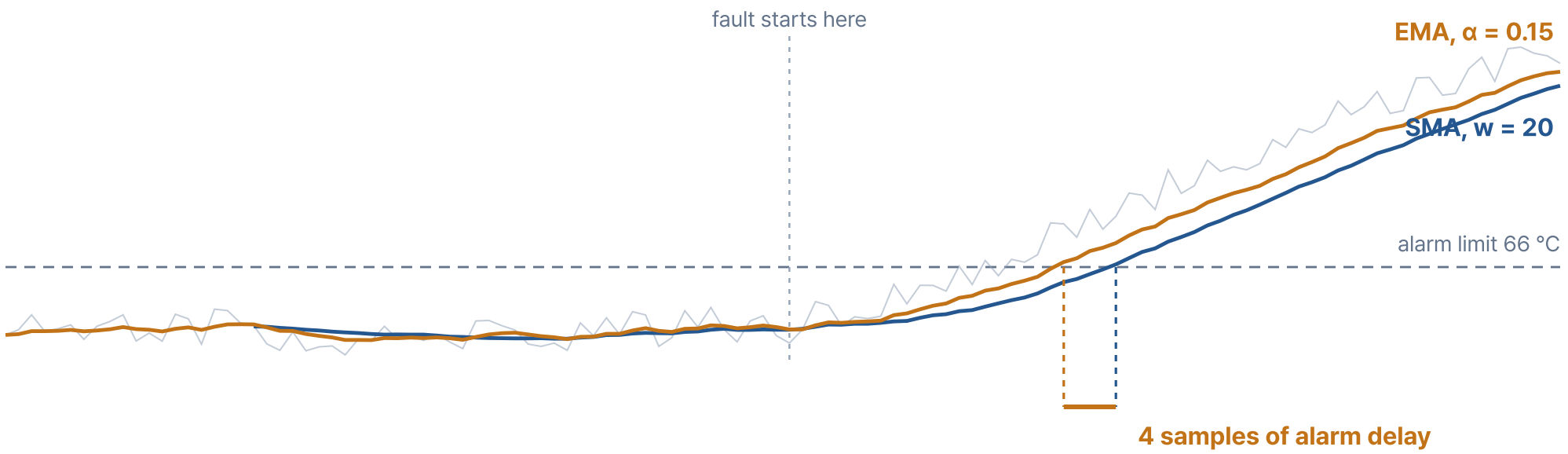
Smoothed

`9, 9, 9, 9, 23, 23, 23, 23, 29, 29, 29, 29`

Twelve values become three distinct ones. **Smoothing buys steadiness with resolution**

Moving Averages for Streams

Simple (SMA) and exponential (EMA) moving averages use only current and past samples



$\hat{x}_t = (1/w) \sum_{k=0}^{w-1} x_{t-k}$ · $s_t = \alpha\, x_t + (1 - \alpha)\, s_{t-1}$ SMA weights the window equally; EMA uses one α

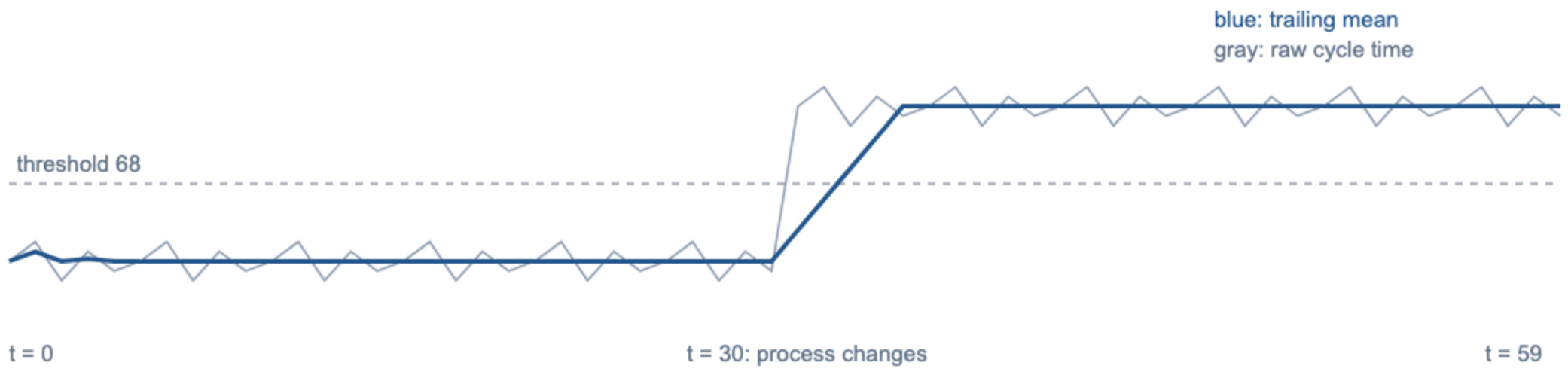
Both use past samples; here EMA crosses four samples before SMA

Demo: Smoothing a Stream

Current and past cycle times (seconds); wider trailing windows respond later

1. Compare windows 1, 5 and 11. 2. When does the smoothed signal cross 68 s?

Window 5



`window 5 · first mean ≥ 68 at t = 32 · delay 2 samples`

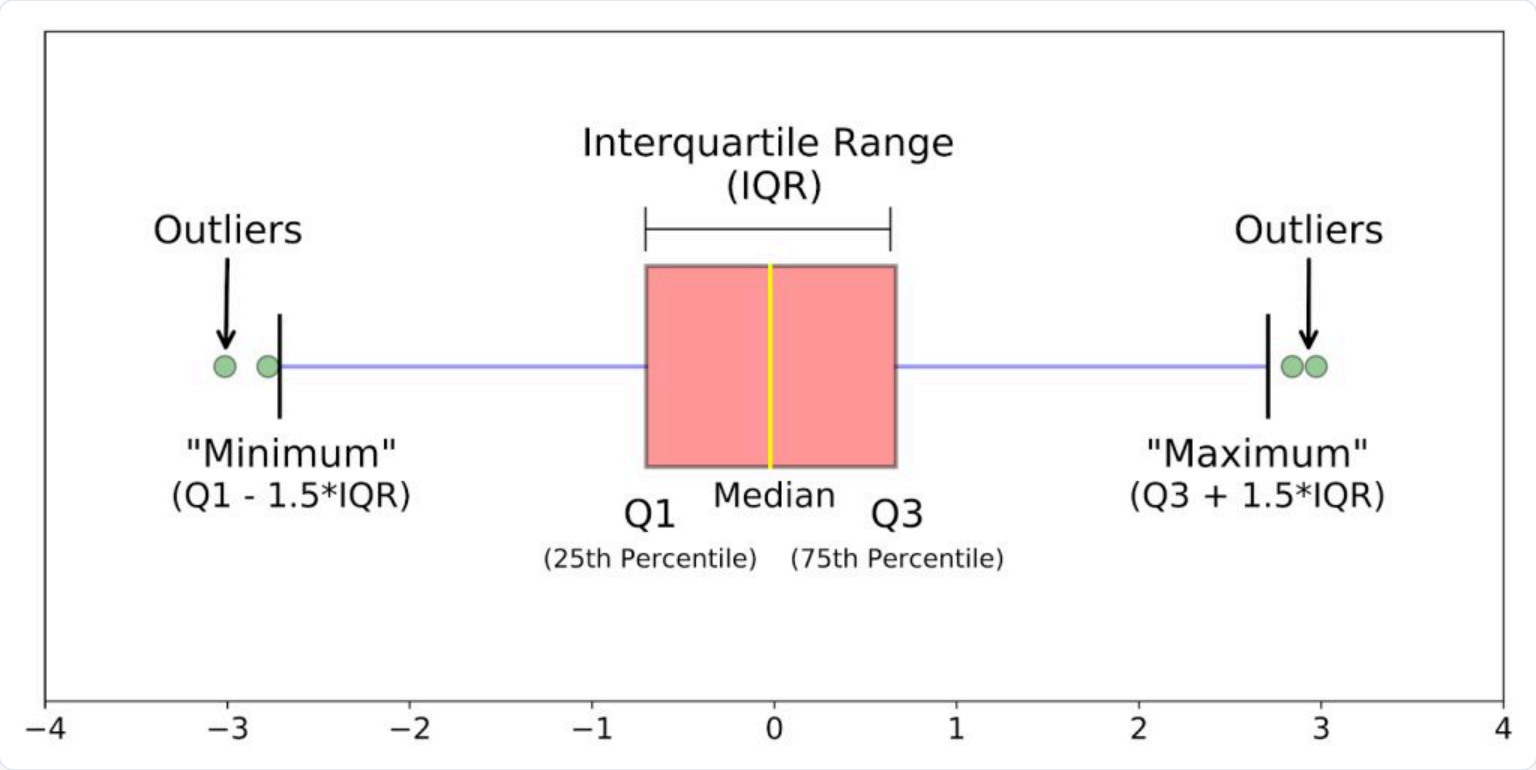
Answer: Smoothing Delay

A larger trailing window needs more new observations

Window	First smoothed value ≥ 68 s	Delay after t = 30
1	30	0 samples
5	32	2 samples
11	35	5 samples

Less fluctuation comes with a later response

Outlier Detection by Statistics



$z_i = (x_i - \bar{x}) / s$ or outside $Q_1 - 1.5 \cdot IQR$, $Q_3 + 1.5 \cdot IQR$ the z-score uses the mean; the fence uses only quartiles

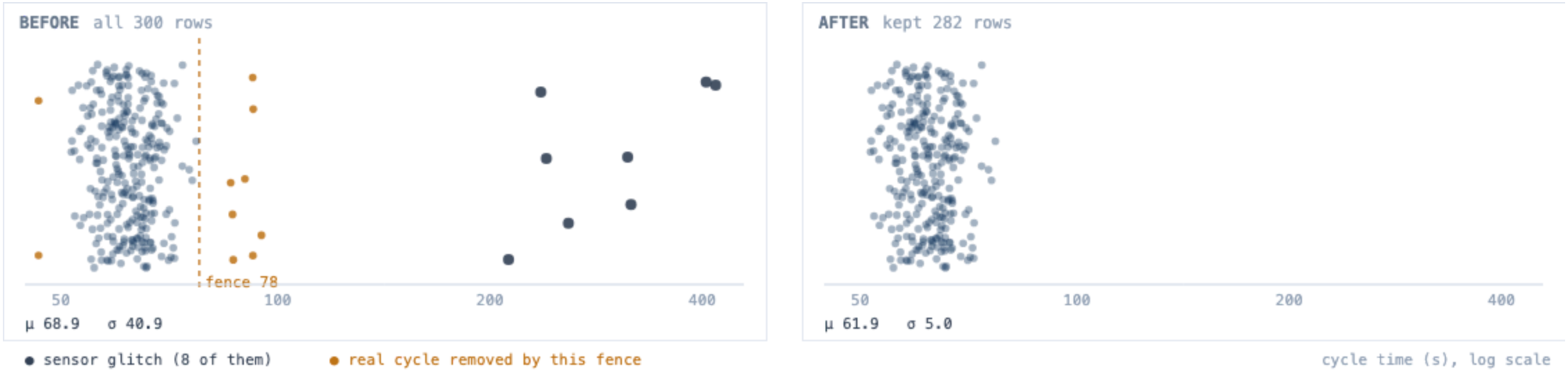
Separate column fences can miss an unlikely combination of otherwise ordinary values

Demo: IQR Fence Width

300 cycle times with eight sensor glitches and eight genuinely slow cycles

1. Change the fence width. 2. Which genuine cycles disappear along with sensor glitches?

Fence width k 1.5 drop anything outside Q1 − k·IQR … Q3 + k·IQR



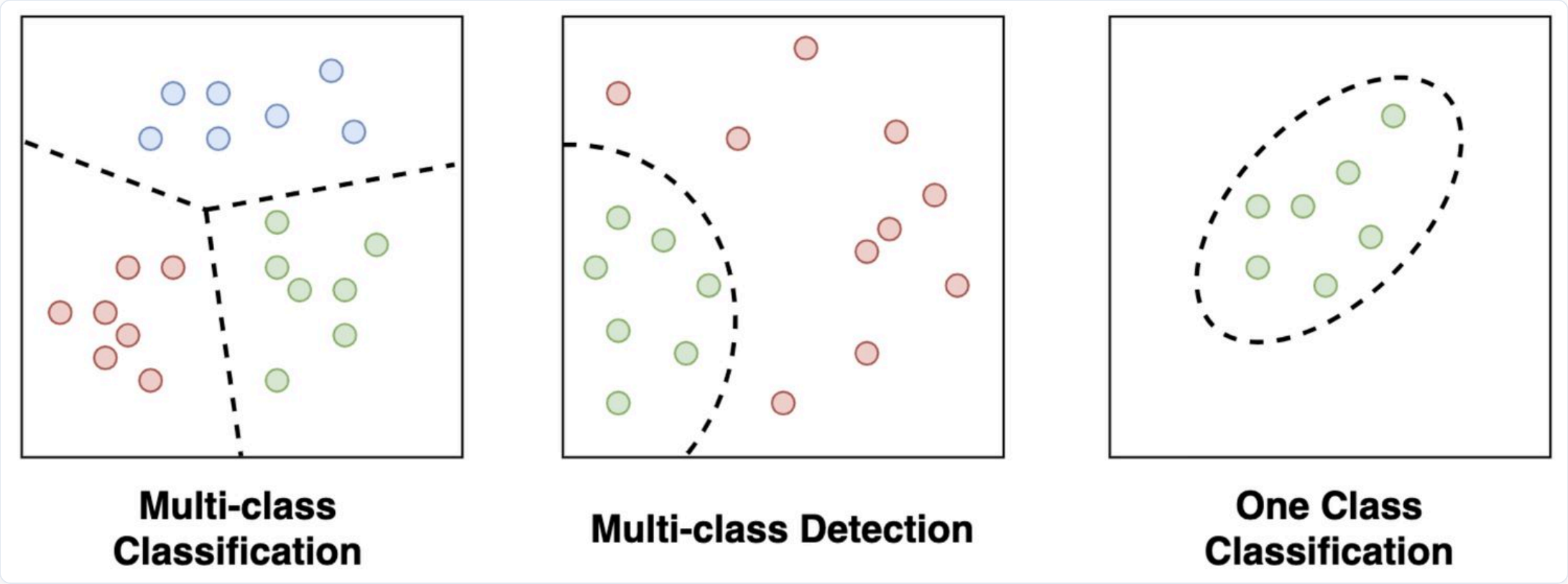
fence at Q3 + 1.5 × IQR = **78 s** · removed **18** rows · σ **40.9 → 5.0**

1.5 is Tukey's convention, not a test. Pick k by what the fence deletes, and report the k you used.

10 real cycles deleted with them

One-Class Classification

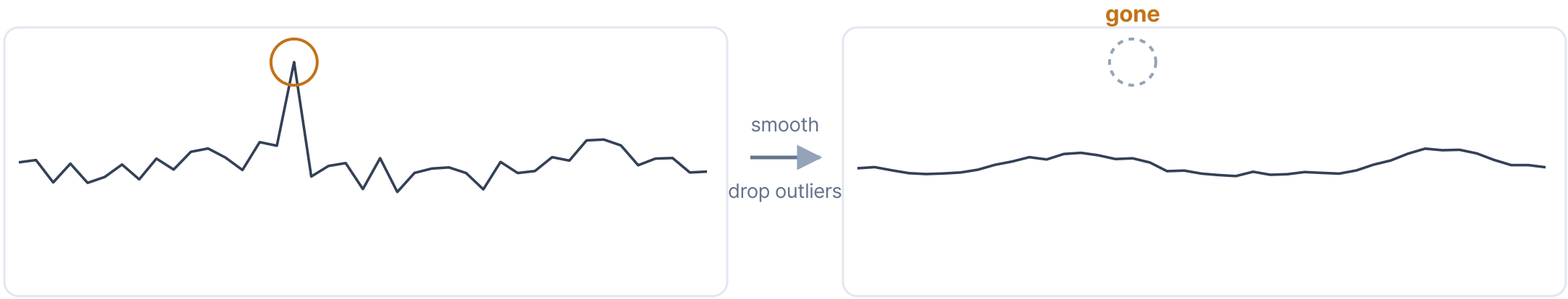
Draw a boundary around normal and flag what falls outside



A learned boundary checks combinations that separate column fences miss

Caveat: Irreversible Cleaning

Cleaning cannot be undone downstream



As recorded

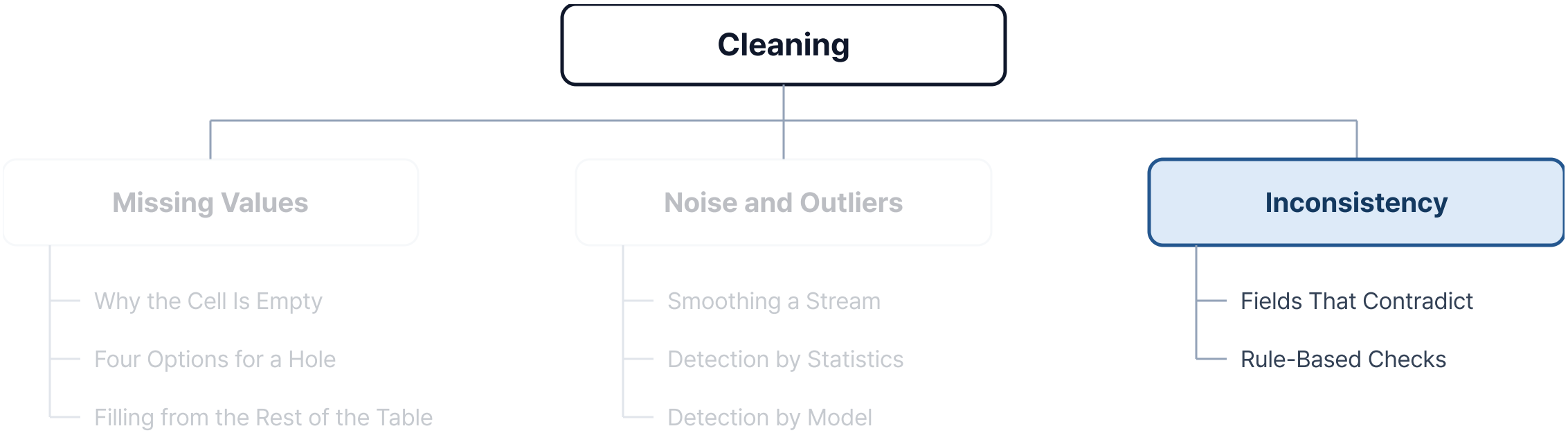
one spike at 14:02

As stored

nothing marks where it was

Deleting the spike removes the evidence needed to revisit the decision

Inconsistency



Fields That Contradict Each Other

ORDER	ORDERED	SHIPPED	QTY	UNIT	AMOUNT
A-1042	2026-03-05	**2026-03-01**	12	kg	480,000
A-1043	2026-03-06	2026-03-09	**0**	kg	**1,250,000**
A-1044	2026-03-06	2026-03-08	26	**lb**	1,040,000
A-1045	2026-03-07	2026-03-10	15	kg	600,000

Nothing is missing here. Each value is wrong only next to another value

Rule-Based Checks

Domain knowledge determines which consistency rules are valid

CHECK	ROW IT STOPS	WHERE THE RULE COMES FROM
shipped ≥ ordered	A-1042	How the business works
qty = 0 → amount = 0	A-1043	How the business works
unit ∈ { kg }	A-1044	A decision someone has to make

Three rules flag three rows; verify the recorded values before training

Demo: Checking Consistency

Apply two domain rules to the recorded fields

run_id	started	ended	total	defects
R-01	09:00	09:20	100	4
R-02	09:30	09:10	80	2
R-03	10:00	10:15	20	24

1. Which rows violate time or count rules?
2. Can the rule tell you which cell is wrong?

Answer: Violations and Repairs

A failed check identifies a contradiction, not the correct replacement

run_id	Violated rule	Action
R-01	none	keep
R-02	ended ≥ started	check source timestamps
R-03	0 ≤ defects ≤ total	check count definitions

Flag contradictions before overwriting recorded values

Integration

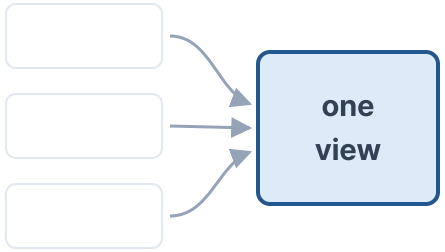
01

Cleaning

Wrong values in one table

02

Integration



Tables that disagree

03

Transformation



Values a model can read

04

Reduction

Less data, same signal

One Customer, Three Systems

Customer relationship management (CRM), supply chain management (SCM), and enterprise resource planning (ERP)

SYSTEM	OWNED BY	KEY	NAME AS STORED	REVENUE FIELD
CRM	Sales	C-10482	Hanil Precision Co., Ltd.	booked_amount
SCM	Logistics	V0000734	HANIL PRECISION	(none)
ERP	Finance	2010-HP	한일정밀(주)	net_sales

No shared key, no shared spelling. Nothing joins these three rows on its own

Three Questions of Integration

Integration

step 1
Schema Mapping

- Which Column Means What
- Field Meanings and IDs

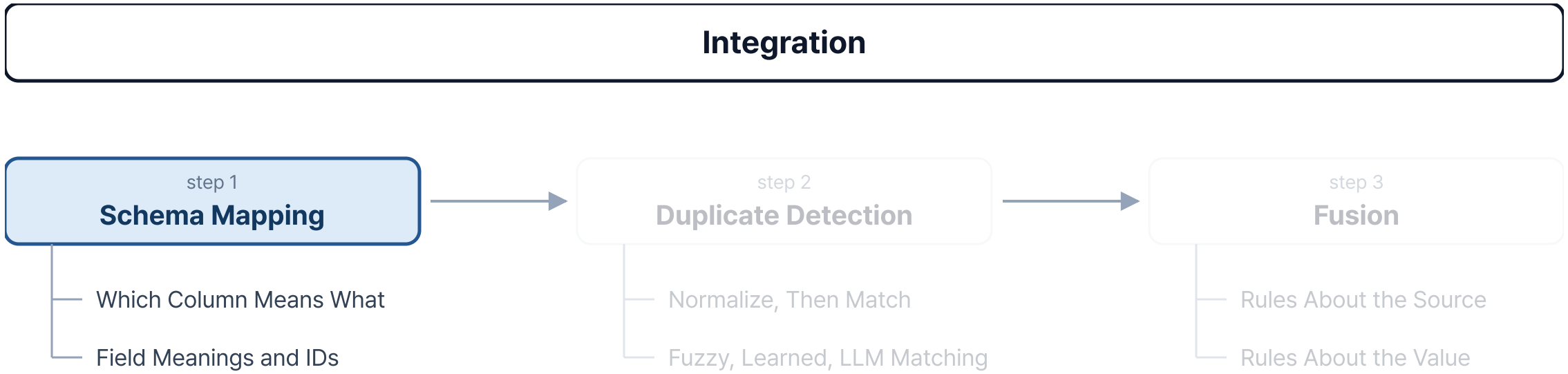
step 2
Duplicate Detection

- Normalize, Then Match
- Fuzzy, Learned, LLM Matching

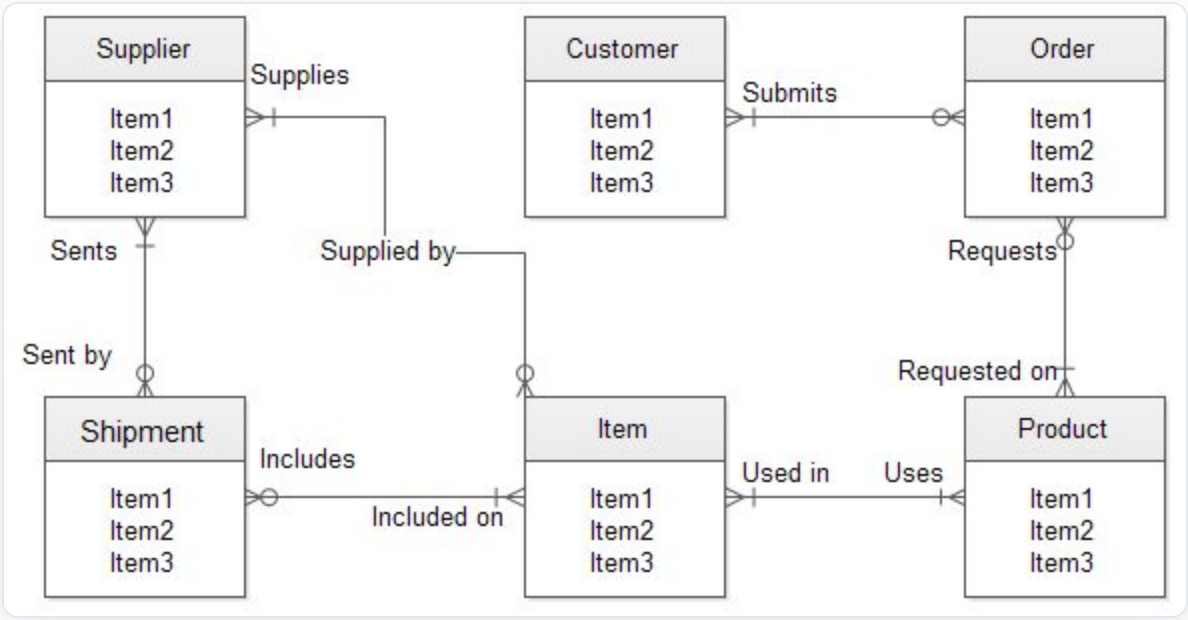
step 3
Fusion

- Rules About the Source
- Rules About the Value

Schema Mapping

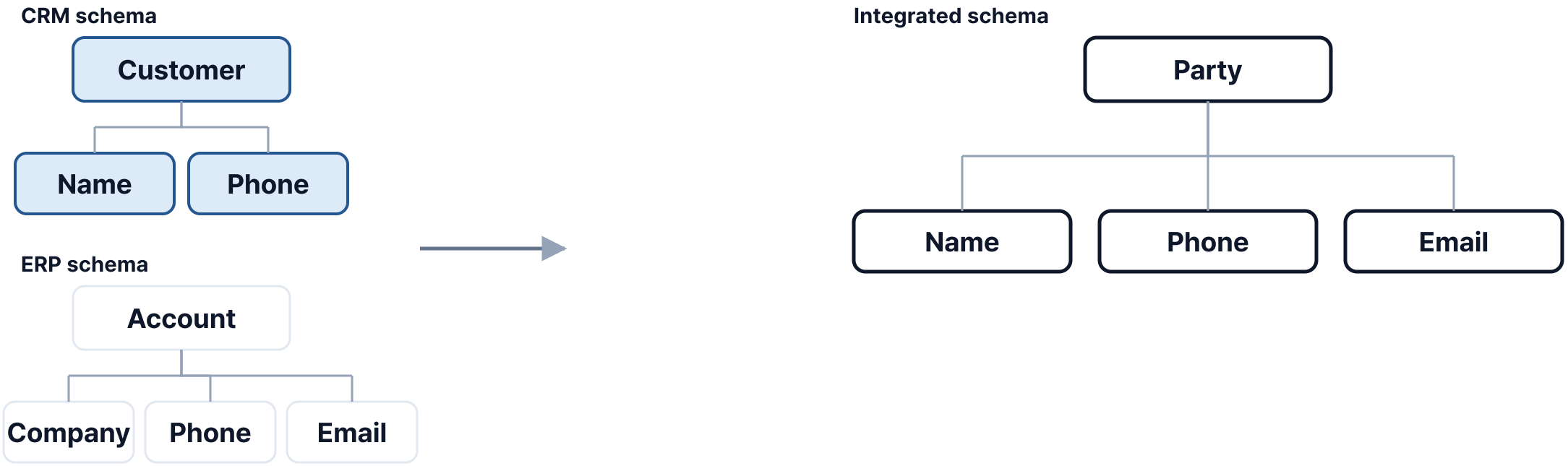


Same Name, Different Entity



Order and Shipment are separate entities here; their names alone do not define the schema

Mapping Two Schemas into One



Matched elements collapse into one. **What only one side has still has to fit somewhere**

Demo: Mapping Customer Records

Match meaning before matching rows

Source field	Target field	Transform or decision
`CRM.Name / ERP.Company`	`Party.name`	`retain name variants`
`CRM.customer_id / ERP.account_id`	`source_id`	`keep source + original ID`
`CRM.booked_amount / ERP.net_sales`	`separate`	`different measures`

1. Do booked_amount and net_sales belong in one column?
2. Do matching column names prove matching customers?

Answer: A Shared Schema

Align field meanings while preserving different measures and source identifiers

Target field	Source mapping	Meaning
`name + aliases`	`CRM.Name / ERP.Company`	`organization names`
`source / source_id`	`CRM ID / SCM ID / ERP ID`	`original identifiers`
`booked_amount`	`CRM.booked_amount`	`booked orders`
`net_sales`	`ERP.net_sales`	`recognized net sales`

Mapping columns does not establish that two rows refer to the same entity

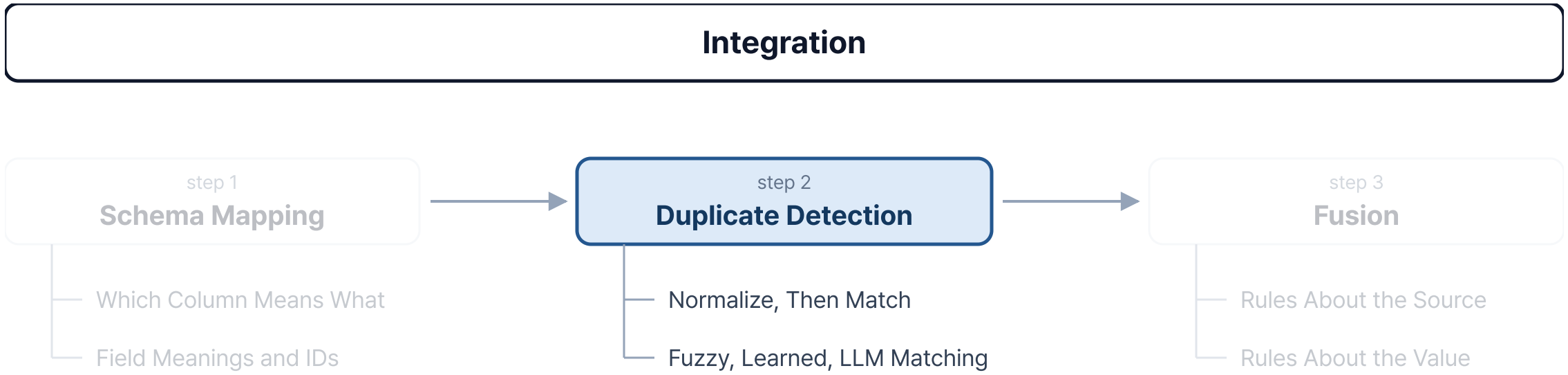
Mapping Fields and Preserving IDs

Align field meanings now; establish shared identity after record matching

CRM field	ERP field	Integrated field
Customer.Name	Account.Company	Party.name
Customer.Phone	Account.Phone	Party.phone
—	Account.Email	Party.email
customer_id	account_id	source + source_id

Matching column meanings does not prove that two records describe the same party

Duplicate Detection



Demo: Exact Duplicates

Normalize, then drop_duplicates: twelve supplier rows, six real suppliers

1. Normalize until six suppliers remain. 2. What breaks when you keep only the first word?

☐ lowercase ☐ strip punctuation ☐ collapse spaces ☐ drop Co / Ltd / Inc ☐ keep the first word `12 rows → 12 · still 6 too many`

BEFORE: 12 ROWS AS STORED	AFTER: THE KEY IT NORMALIZES TO	KEPT
Samsung Electronics Co., Ltd.	`Samsung Electronics Co., Ltd.`	**keep**
samsung electronics co ltd	`samsung electronics co ltd`	**keep**
Samsung Electronics Co., Ltd.	`Samsung  Electronics Co., Ltd.`	**keep**
Samsung Electronics	`Samsung Electronics`	**keep**
LG Electronics Inc.	`LG Electronics Inc.`	**keep**
LG ELECTRONICS INC.	`LG ELECTRONICS INC.`	**keep**
LG Electronics	`LG Electronics`	**keep**
Hyundai Mobis Co., Ltd.	`Hyundai Mobis Co., Ltd.`	**keep**
Hyundai Mobis	`Hyundai Mobis`	**keep**
POSCO Holdings	`POSCO Holdings`	**keep**
POSCO International Corp.	`POSCO International Corp.`	**keep**
Samsung Heavy Industries Co., Ltd.	`Samsung Heavy Industries Co., Ltd.`	**keep**

Normalize before you dedup, never after. Casing, whitespace and punctuation account for most byte-level mismatches.

Answer: Normalization and Merging

A shorter string can accidentally erase a real distinction

Normalization	Result in this example
Case, punctuation, spaces, suffixes	12 rows → 6 suppliers
Keep only the first word	Distinct Samsung and POSCO entities merge
Preserve original name and source ID	The decision remains auditable

Normalization should support matching without hiding the original record

Candidates and Confirmed Matches

Similarity proposes record pairs; evidence supports a shared identity

Schema already aligned

```
name · phone ·
source_id
```

→

Candidate pair

```
Hanil Precision Co.,
Ltd.
Hanil Precision Co Ltd

Similar names suggest
a match
```

→

Identity evidence

```
Verified registration
ID

Confirm → one party_id
Unclear → keep for
review
```

Preserve source IDs while deciding which records refer to one entity

Fuzzy Matching

Each field gets a similarity score, then one weighted total against a cutoff



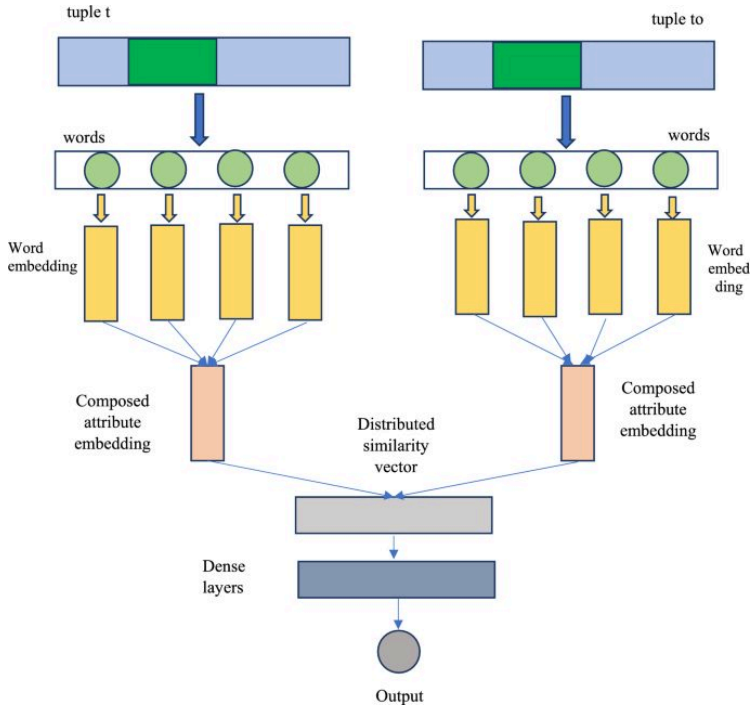
$$\text{score} = \sum_f w_f \cdot \text{sim}_f(a, b) \qquad \text{sim} = 1 - \text{edit}(a,b) \,/\, \max(|a|,|b|)$$

on lowercased text, then one weighted total

Per-field scores become one number. **The cutoff you pick trades false merges against missed ones**

Learned Matching

Learned representations compare records beyond exact spelling



Input

Two candidate records

Model

Encode fields and estimate similarity

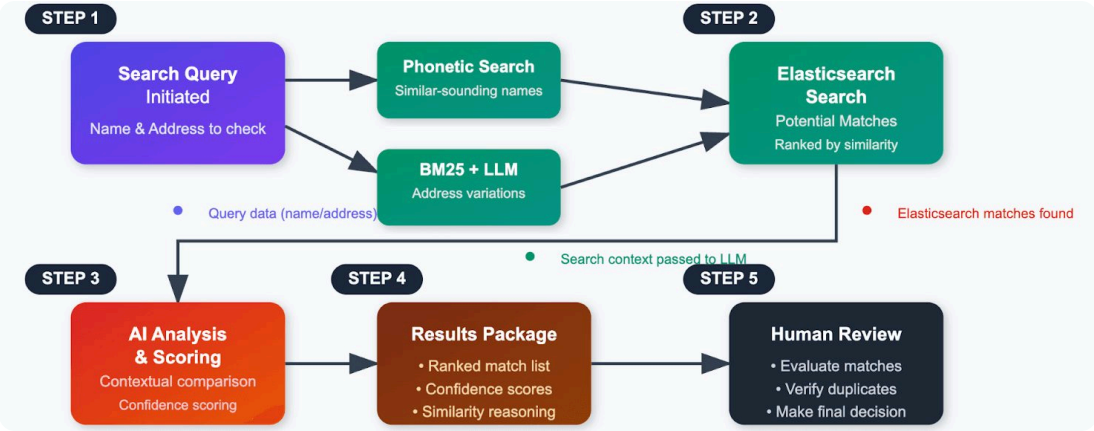
Decision

A score proposes a match; evidence confirms it

Learned similarity can link different spellings; identity still needs evidence

LLM-Assisted Matching

Large language models (LLMs) propose the matches, humans verify them



Input

Candidate names and record fields

Model

Retrieve evidence and propose a match

Review

Verify identity before merging records

Models propose candidates; verified identity supports a merge

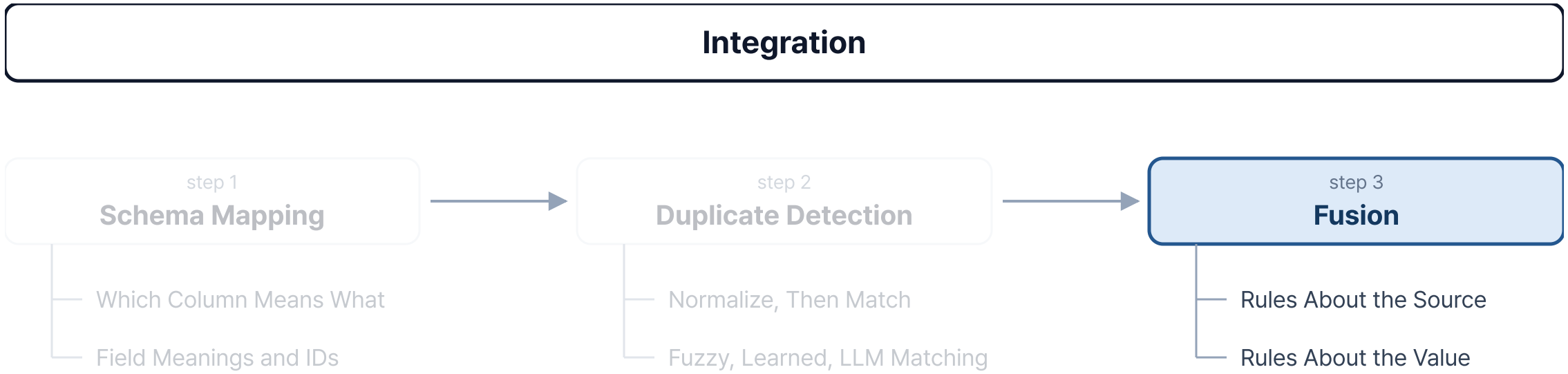
Answer: Source IDs and Party IDs

A verified link groups records without rewriting their original identifiers

source	source_id	party_id	basis
CRM	C-10482	P-073	verified registration ID
SCM	V0000734	P-073	verified registration ID
ERP	2010-HP	P-073	verified registration ID

Similar names generate candidates; verified evidence supports a merge

Fusion


Integration
step 1
Schema Mapping
Which Column Means What
Field Meanings and IDs
step 2
Duplicate Detection
Normalize, Then Match
Fuzzy, Learned, LLM Matching
step 3
Fusion
Rules About the Source
Rules About the Value

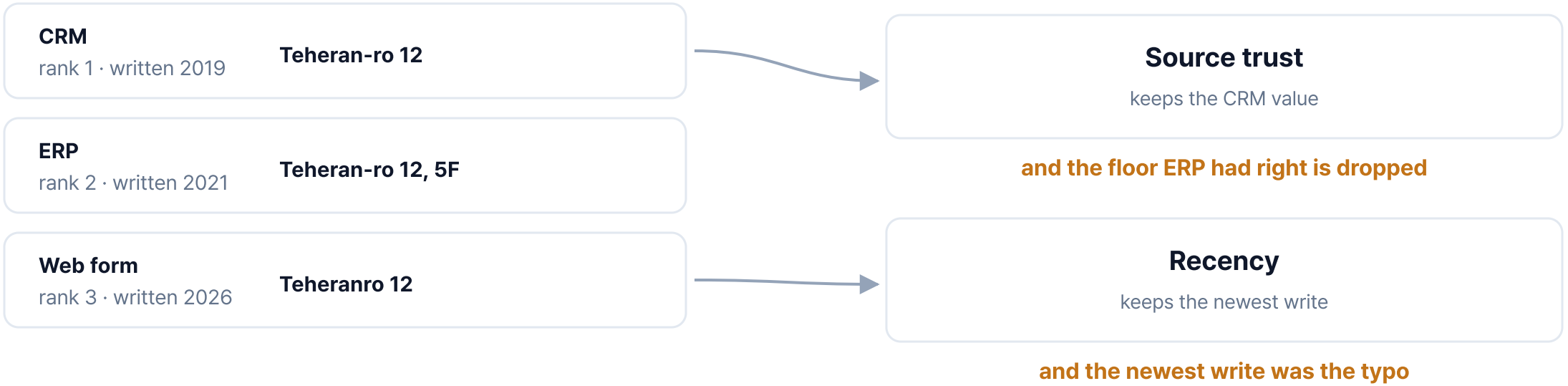
Conflicts After Matching

Matched rows still disagree on values

	name	address	phone
CRM	Hanil Precision	Teheran-ro 12	02-555-0100
SCM	Hanil Precision	Teheran-ro 12	02-555-0100
ERP	Hanil Precision	Teheran-ro 12, 5F	02-555-0199
Web form	Hanil Precision	Teheranro 12	010-2233-4455

Same customer, three spellings of one address, three phone numbers

Rules About the Source



Neither rule reads the value. **One keeps a stale address, the other keeps a typo**

Rules About the Value

Majority voting

CRM	**02-555-0100**
copy	**02-555-0100**
report	**02-555-0100**
ERP	**02-555-0199**

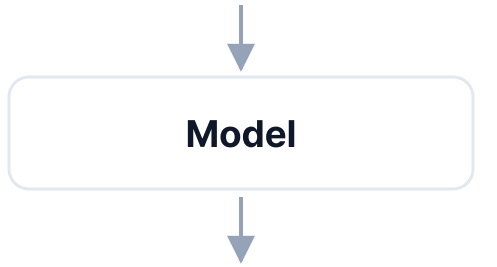
three agree, but two of them are copies of the first

Learned

conflicts a person already settled

Seoul vs SEOUL → a person kept Seoul

02-555-0100 vs 0199 → a person kept 0100



picks the winner of the next conflict

$$\hat{x} = \operatorname{argmax}_v \sum_k w_k \cdot 1[x_k = v]$$ weighted vote: w_k is the trust in source k

Three copies of one value **count as one vote, not three**

Demo: Resolving a Customer Conflict

Choose a canonical value and keep the evidence

Field	Candidate values	Evidence
name	Hanil Precision / 한일정밀(주)	same verified registration ID
address	12 / 12, 5F / Teheranro 12	verified delivery: 12, 5F
phone	02-555-0100 / 02-555-0199 / 010-2233-4455	no fresh confirmation

1. Which values can be settled now?
2. Which field needs a review instead of a vote?

Answer: A Canonical Customer

Publish resolved values while keeping unresolved conflicts visible

party_id	field	canonical value	evidence / status
P-073	name	Hanil Precision	verified ID; aliases retained
P-073	address	Teheran-ro 12, 5F	verified delivery + ERP
P-073	phone	NULL	review; candidates retained

A canonical record includes provenance and unresolved decisions

Takeaway: Cleaning and Integration

Decision	What must remain clear
Repair a value	Observed, estimated or verified against the source
Flag an error	A contradiction identifies a problem, not its replacement
Merge records	Field meaning, entity identity and conflict resolution

Preserve original evidence and record why each value or identity changed