CS 4440 A

Emerging Database Technologies

Lecture 17

04/15/24

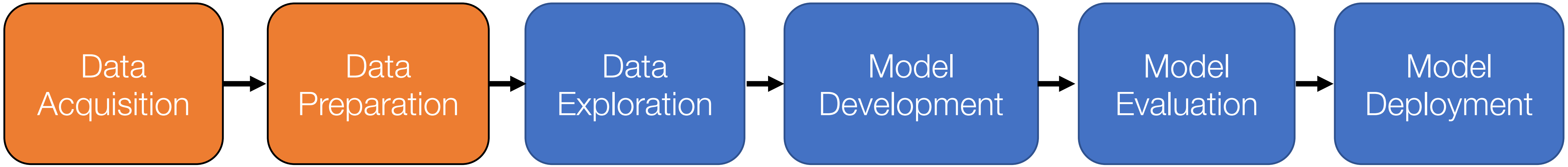
Announcements

- Paper Critique
  - Due Wednesday

- Project Presentation
  - Send your slides to Catherine by 2PM on the day of presentation

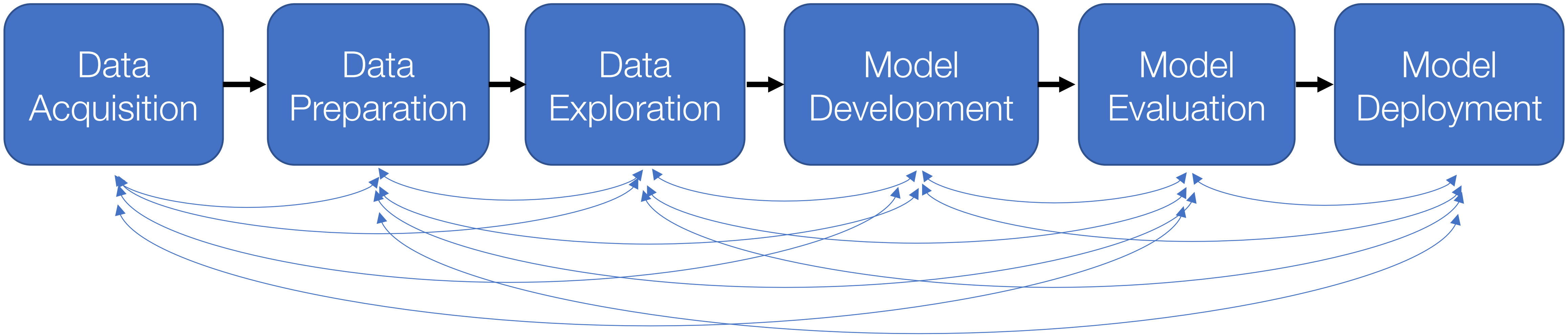
- Project Demo
  - April 26, up to 15min per group

Data Curation Challenges in the ML Lifecycle



ML lifecycle in a bird's eye view

“Only a fraction of real-world ML systems is composed of ML code” [1]



The machine learning lifecycle is complex and iterative process

Humans play an important role in almost all steps of the lifecycle

[1] Sculley, David, et al. "Hidden technical debt in machine learning systems." NeurIPS 2015

Human roles in data analytics

Data Acquisition → Data Preparation → Data Exploration → Model Development → Model Evaluation → Model Deployment

↑ Data Acquisition ↑ Data Preparation

amazon mechanical turk beta

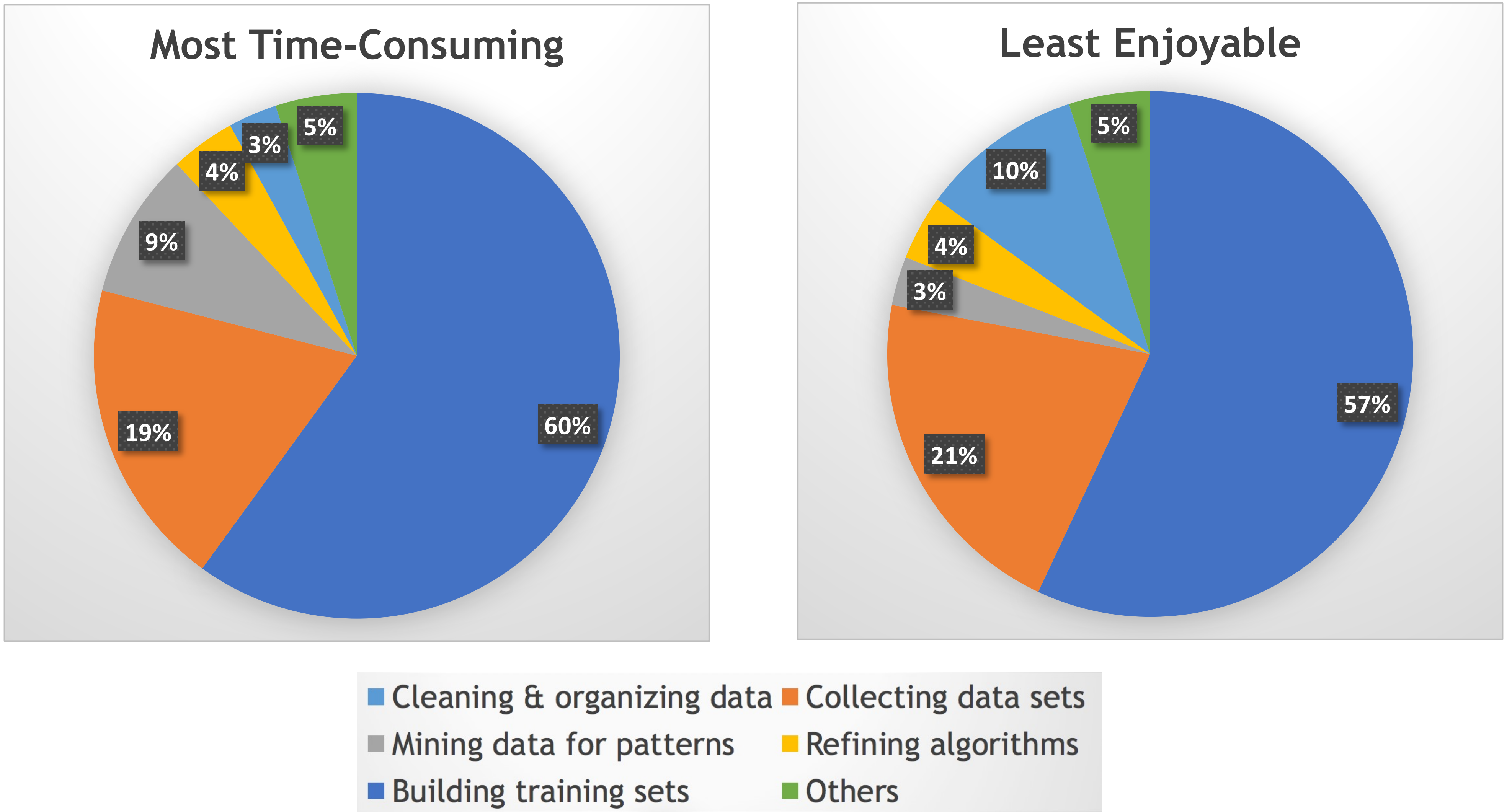
TRIFACTA

Data Producer

Humans play an important role in almost all steps of the lifecycle

Cleaning Data: Most Time-Consuming, Least Enjoyable Data Science Task

Forbes, 2016



Common Data Errors

Incomplete

Country	UN R/P 10%[4]	UN R/P 20%[5]	World Bank Gini (%)[6]	WB Gini (year)	CIA R/P 10%[7]	Year	CIA Gini (%)[8]	CIA Gini (year)	GPI Gini (%)[9]
Seychelles			65.8	2007					
Comoros			64.3	2004					
Namibia	106.6	56.1	63.9	2004	129.0	2003	59.7	2010	
South Africa	33.1	17.9	63.1	2009	31.9	2000	65.0	2005	
Botswana	43.0	20.4	61.0	1994			63	1993	
Haiti	54.4	26.6	59.2	2001	68.1	2001	59.2	2001	
Angola			58.6	2000					62.0
Honduras	59.4	17.2	57.0	2009	35.2	2003	57.7	2007	

Adapted from Intro to Data Cleaning lecture from Xu Chu

Common Data Errors

Inaccurate

Common Data Errors

Inconsistent

FlightView

American Airlines Flight Number 119 (AA119)

FLIGHT TRACKER

Departure

Airport:	
Scheduled Time:	6:15 PM, Dec 08
Takeoff Time:	6:53 PM, Dec 08
Terminal - Gate:	Terminal A - 32

ArrivalStatus: In Air

Airport:	
Scheduled Time:	9:40 PM, Dec 08
	9:42 PM, Dec 08
Estimated Time:	Track This Flight Live!
Time Remaining:	25 min
Terminal - Gate:	Terminal 4 - 42B
Baggage Claim:	4

FlightAware

AAL119 (Track inbound flight)
(web site) (all flights)
American Airlines "American"

Aircraft	Boeing 737-800 (twin-jet) (B738/Q - track or photos)
Origin	Terminal A / Gate 32 / Newark Liberty Intl (KEWR - tr
Destination	Terminal 4 / Gate 42B / Los Angeles Intl (KLAX - track
	Other flights between these airports
Route	ZIMMZ Q42 BTRIX Q480 AIR J80 VHP J80 MCI J24 SLN J102 ALS J44 RSK J (Decode)
Date	2011年 12月 08日 (Thursday)
Duration	5 hours 43 minutes
Progress	20 minutes left 5 hours 23 minutes
Status	En Route (2,284 sm down; 168 sm to go)
Dis	Direct: 2,451 sm Planned: 2,458
Fare	$51.99 to $3,561.11; average: $241.96 (airline insight)
Cabin	First: Dinner / Economy: Food for sale

	Scheduled	7-day Average	*Actual/Estimated*
Departure	06:15PM EST	07:08PM EST	06:53PM EST
Arrival	08:33PM PST	09:17PM PST	*09:36PM PST*

Orbitz

American Airlines # 119

Leg 1: In Transit

Departs: Newark (EWR) View real-time airpo

Gate: 32

Scheduled	Estimated	Actual
6:22p Dec 8	-	6:32p Dec 8

Arrives: Los Angeles (LAX) View real-time ai

Gate: 42B

Scheduled	Estimated	Actual
9:54p Dec 8	9:47p Dec 8	

Adapted from Intro to Data Cleaning lecture from Xu Chu

Common Data Errors

Duplicated

Merged citations

This "Cited by" count includes citations to the following articles in Scholar. The ones marked * may be different from the article in the profile.

Scaling a Declarative Cluster Manager Architecture with Query Optimization Techniques (Technical Report) 1 *
K Rong, M Budiu, A Skiadopoulos, L Suresh, A Tai
2022

Scaling a Declarative Cluster Manager Architecture with Query Optimization Techniques
K Rong, M Budiu, A Skiadopoulos, L Suresh, A Tai
Proceedings of the VLDB Endowment 16 (10), 2618-2631, 2023

Adapted from Intro to Data Cleaning lecture from Xu Chu

Data Quality Rules

	Name	ID	LVL	ZIP	ST	SAL
t_1	Alice	ID1	5	10001	NM	90K
t_2	Bob	ID2	6	87101	NM	80K
t_3	Chris	ID3	4	10001	NY	80K
t_4	Dave	ID4	1	90057	CA	20K
t_5	Frank	ID5		90057	CA	50K

R1: Two persons with the same ZIP live in the same ST

Data Quality Rules

	Name	ID	LVL	ZIP	ST	SAL
t_1	Alice	ID1	5	10001	NM	90K
t_2	Bob	ID2	6	87101	NM	80K
t_3	Chris	ID3	4	10001	NY	80K
t_4	Dave	ID4	1	90057	CA	20K
t_5	Frank	ID5		90057	CA	50K

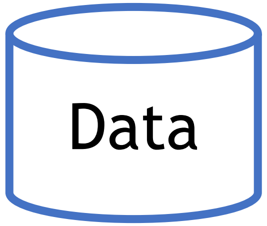
R2: LVL should not be empty

Data Quality Rules

	Name	ID	LVL	ZIP	ST	SAL
t_1	Alice	ID1	5	10001	NM	90K
t_2	Bob	ID2	6	87101	NM	80K
t_3	Chris	ID3	4	10001	NY	80K
t_4	Dave	ID4	1	90057	CA	20K
t_5	Frank	ID5		90057	CA	50K

R3: People with a higher LVL earn more SAL in the same ST

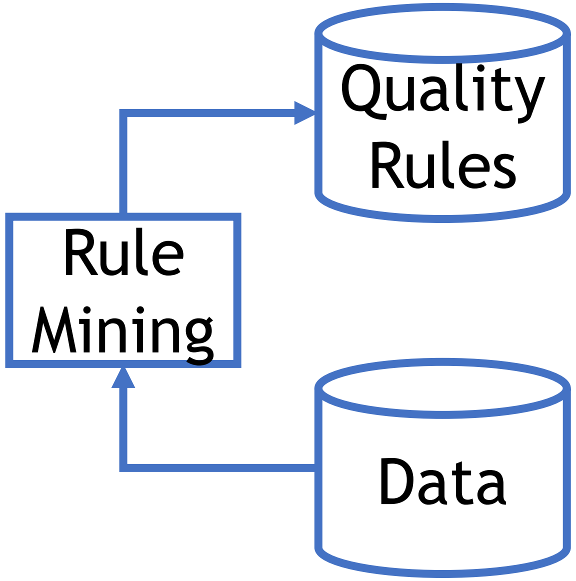
Rule-based Data Cleaning



Name	ZIP	ST
Alice	10001	NM
Bob	87101	NM
Chris	10001	NY

Adapted from Intro to Data Cleaning lecture from Xu Chu

Rule-based Data Cleaning

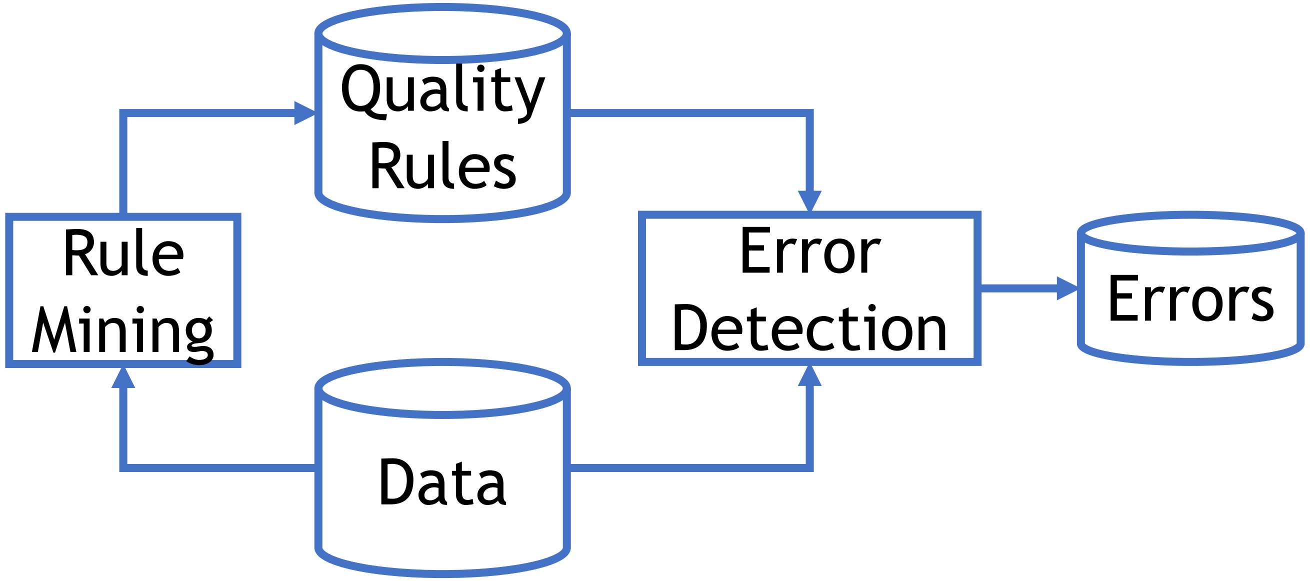


Name	ZIP	ST
Alice	10001	NM
Bob	87101	NM
Chris	10001	NY

Two persons with the same ZIP live in the same ST

Adapted from Intro to Data Cleaning lecture from Xu Chu

Rule-based Data Cleaning

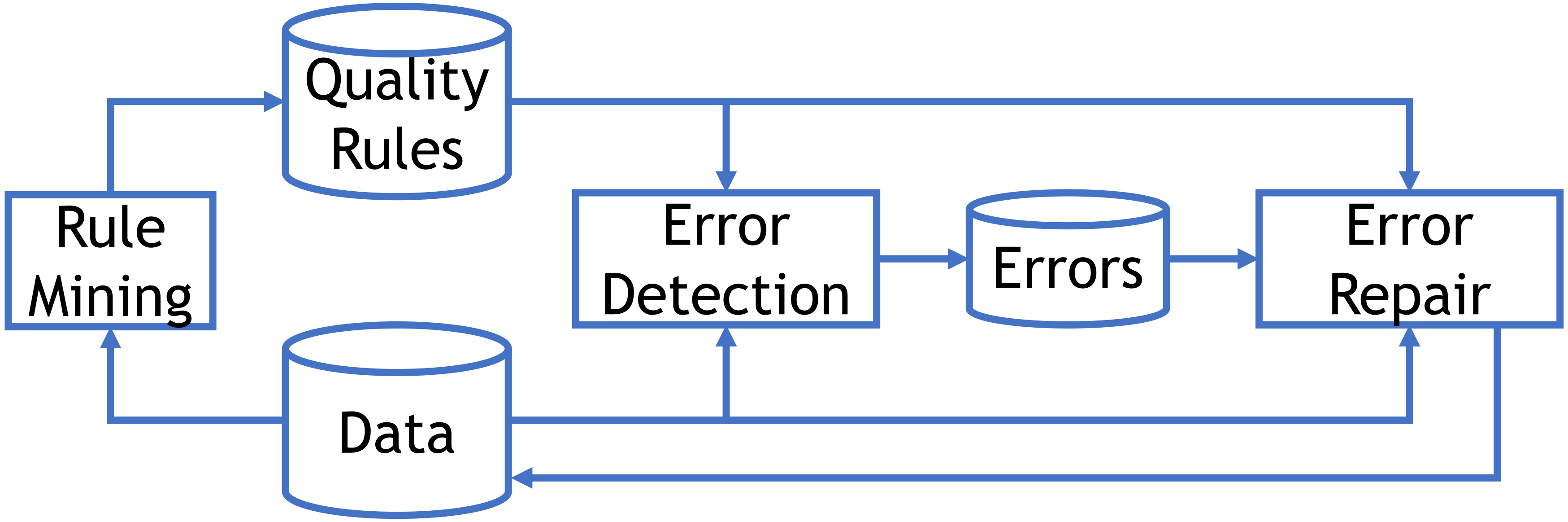


Name	ZIP	ST
Alice	10001	NM
Bob	87101	NM
Chris	10001	NY

Two persons with the same ZIP live in the same ST

Adapted from Intro to Data Cleaning lecture from Xu Chu

Rule-based Data Cleaning

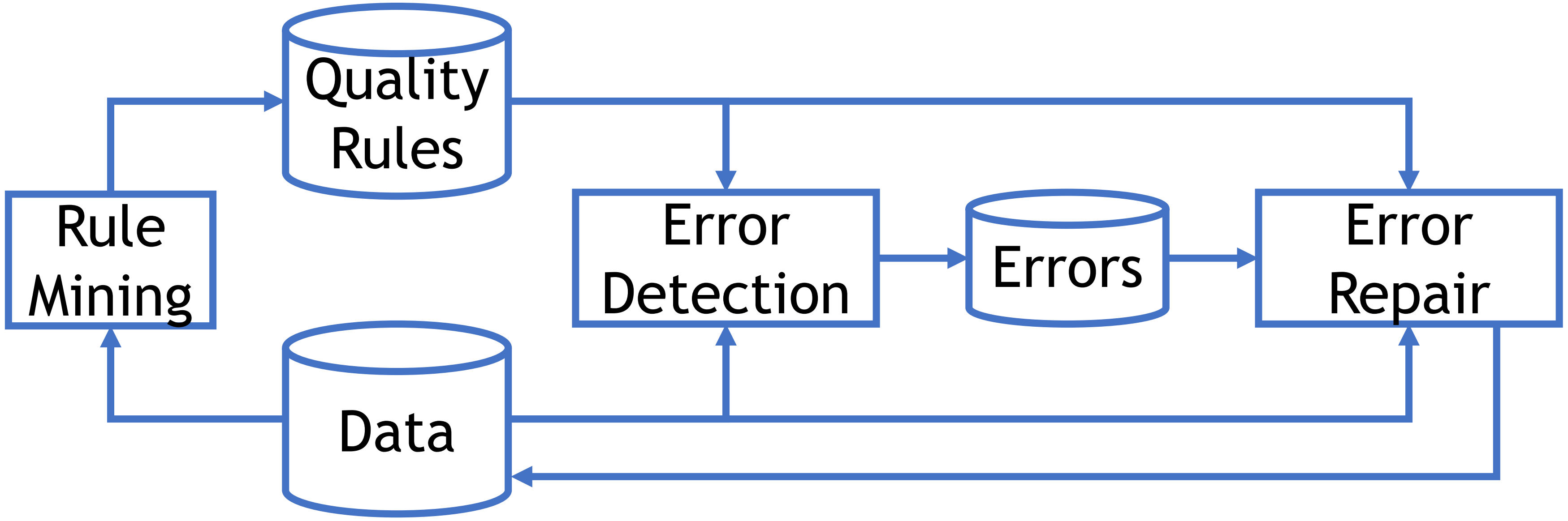


Name	ZIP	ST
Alice	10001	NY
Bob	87101	NM
Chris	10001	NY

Two persons with the same ZIP live in the same ST

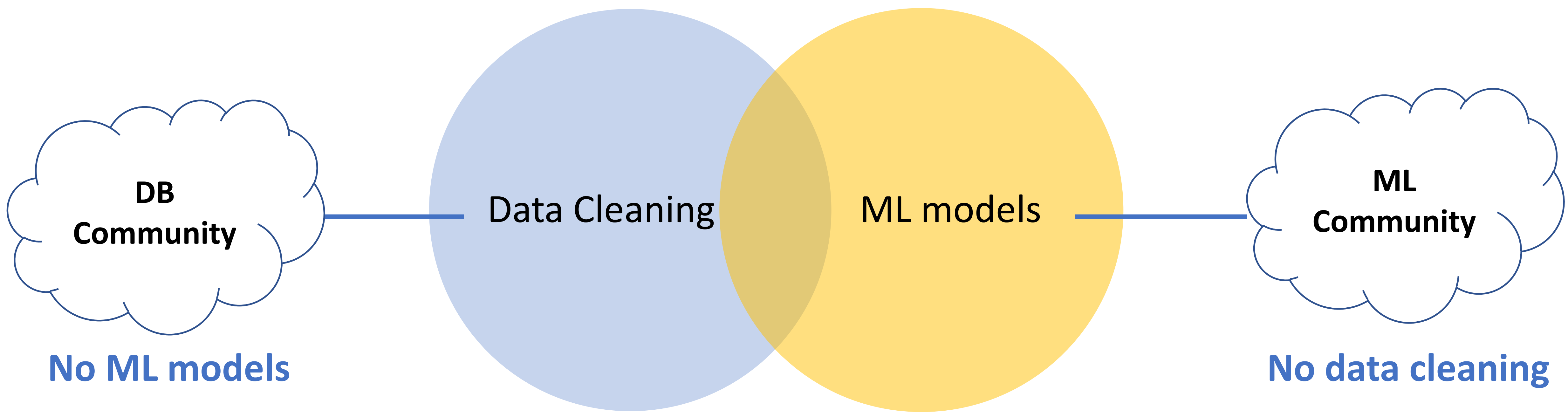
Adapted from Intro to Data Cleaning lecture from Xu Chu

Two tasks in data cleaning



- Detection: A minimal set of cells that cannot coexist together
- Repair: A set of cell updates to resolve the violations

Data cleaning and ML



Data cleaning and ML

Cleaning ”before” ML:

- Perform cleaning independently of the downstream ML applications; leverage user-specified signals or data-driven approaches
- Example: HoloClean: Holistic Data Repairs with Probabilistic Inference
  - Also an example of using ML for data cleaning

Reading: From Cleaning Before ML to Cleaning For ML

HoloClean: Holistic Data Repairs with Probabilistic Inference. [VLDB'17]

Input

Dataset to be cleaned

	DBAName	Address	City	State	Zip
t1	John Veliotis Sr.	3465 S Morgan ST	Chicago	IL	60608
t2	John Veliotis Sr.	3465 S Morgan ST	Chicago	IL	60609
t3	John Veliotis Sr.	3465 S Morgan ST	Chicago	IL	60609
t4	Johnnyo's	3465 S Morgan ST	Cicago	IL	60608

Denial Constraints

c1: DBAName $\rightarrow$ Zip
c2: Zip $\rightarrow$ City, State
c3: City, State, Address $\rightarrow$ Zip

Matching Dependencies

m1: $Zip = Ext\_Zip \rightarrow City = Ext\_City$
m2: $Zip = Ext\_Zip \rightarrow State = Ext\_State$
m3: $City = Ext\_City \wedge State = Ext\_State \wedge$
$\wedge\ Address = Ext\_Address \rightarrow Zip = Ext\_Zip$

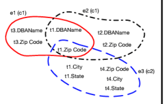
External Information

Ext_Address	Ext_City	Ext_State	Ext_Zip
3465 S Morgan ST	Chicago	IL	60608
1208 N Wells ST	Chicago	IL	60610
259 E Erie ST	Chicago	IL	60611
2806 W Cermak Rd	Chicago	IL	60623

The HoloClean Framework

1. Error Detection Module

- Use integrity constraints
- Leverage external data
- Detect outliers
- Identify possible repairs

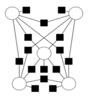


2. Compilation Module

- Automatic Featurization
- Statistical analysis and candidate repair generation
- Compilation to probablistic program

3. Repair Module

- Ground probalistic model
- Statistical learning
- Probabilistic inference



Output

Proposed Cleaned Dataset

	DBAName	Address	City	State	Zip
t1	John Veliotis Sr.	3465 S Morgan ST	Chicago	IL	60608
t2	John Veliotis Sr.	3465 S Morgan ST	Chicago	IL	**60608**
t3	John Veliotis Sr.	3465 S Morgan ST	Chicago	IL	**60608**
t4	**John Veliotis Sr.**	3465 S Morgan ST	**Chicago**	IL	60608

Marginal Distribution of Cell Assignments

Cell	Possible Values	Probability
t2.Zip	60608	0.84
	60609	0.16
t4.City	Chicago	0.95
	Cicago	0.05
t4.DBAName	John Veliotis Sr.	0.99
	Johnnyo's	0.01

Probabilistic model that unifies different signals for repairing a dataset.

Adapted from Intro to Data Cleaning lecture from Prof. Xu Chu

Constraints and minimality

Functional dependencies

$$\begin{aligned}
&c1:\ \text{DBAName} \rightarrow \text{Zip}\\
&c2:\ \text{Zip} \rightarrow \text{City, State}\\
&c3:\ \text{City, State, Address} \rightarrow \text{Zip}
\end{aligned}$$

	DBAName	AKAName	Address	City	State	Zip
t1	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	***Chicago***	IL	***60608***
t2	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	***60609***
t3	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	***60609***
t4	***Johnnyo's***	Johnnyo's	3465 S Morgan ST	***Cicago***	IL	60608

Bohannon et al., 2005, 2007; Kolahi and Lakshmanan , 2005; Bertossi et al., 2011; Chu et al., 2013; 2015 Fagin et al., 2015

Constraints and minimality

Functional dependencies

$c1: \text{DBAName} \rightarrow \text{Zip}$

$c2: \text{Zip} \rightarrow \text{City, State}$

$c3: \text{City, State, Address} \rightarrow \text{Zip}$

	DBAName	AKAName	Address	City	State	Zip
t1	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	**60609**
t2	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	**60609**
t3	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	**60609**
t4	**Johnnyo's**	Johnnyo's	3465 S Morgan ST	**Cicago**	IL	60608

Action: Fewer erroneous than correct cells; perform minimum number of changes to satisfy all constraints

Constraints and minimality

Functional dependencies

c1: DBAName $\rightarrow$ Zip
c2: Zip $\rightarrow$ City, State
c3: City, State, Address $\rightarrow$ Zip

	DBAName	AKAName	Address	City	State	Zip
t1	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	**60609**
t2	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	60609
t3	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	60609
t4	Johnnyo's	Johnnyo's	3465 S Morgan ST	Cicago	IL	60608

Error; correct zip code is 60608

Does not fix errors and introduces new ones.

External Information

Matching dependencies

$$m1: \ Zip = Ext\_Zip \rightarrow City = Ext\_City$$
$$m2: \ Zip = Ext\_Zip \rightarrow State = Ext\_State$$
$$m3: \ City = Ext\_City \wedge State = Ext\_State \wedge$$
$$\wedge \ Address = Ext\_Address \rightarrow Zip = Ext\_Zip$$

External list of addresses

Ext_Address	Ext_City	Ext_State	Ext_Zip
3465 S Morgan ST	Chicago	IL	60608
1208 N Wells ST	Chicago	IL	60610

	DBAName	AKAName	Address	City	State	Zip
t1	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	***Chicago***	IL	***60608***
t2	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	***60609***
t3	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	***60609***
t4	***Johnnyo's***	Johnnyo's	3465 S Morgan ST	***Cicago***	IL	60608

Fan et al., 2009; Bertossi et al., 2010; Chu et al., 2015

External Information

Matching dependencies

$$\text{m1: } \mathrm{Zip} = \mathrm{Ext\_Zip} \rightarrow \mathrm{City} = \mathrm{Ext\_City}$$
$$\text{m2: } \mathrm{Zip} = \mathrm{Ext\_Zip} \rightarrow \mathrm{State} = \mathrm{Ext\_State}$$
$$\text{m3: } \mathrm{City} = \mathrm{Ext\_City} \wedge \mathrm{State} = \mathrm{Ext\_State} \wedge \wedge \mathrm{Address} = \mathrm{Ext\_Address} \rightarrow \mathrm{Zip} = \mathrm{Ext\_Zip}$$

External list of addresses

Ext_Address	Ext_City	Ext_State	Ext_Zip
3465 S Morgan ST	Chicago	IL	60608
1208 N Wells ST	Chicago	IL	60610

	DBAName	AKAName	Address	City	State	Zip
t1	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	60608
t2	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	**60608**
t3	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	**60608**
t4	**Johnnyo's**	Johnnyo's	3465 S Morgan ST	**Chicago**	IL	60608

Action: Map external information to input dataset using matching dependencies and repair disagreements

External Information

Matching dependencies

$$m1:\ Zip = Ext\_Zip \rightarrow City = Ext\_City$$
$$m2:\ Zip = Ext\_Zip \rightarrow State = Ext\_State$$
$$m3:\ City = Ext\_City \wedge State = Ext\_State \wedge$$
$$\wedge\ Address = Ext\_Address \rightarrow Zip = Ext\_Zip$$

External list of addresses

Ext_Address	Ext_City	Ext_State	Ext_Zip
3465 S Morgan ST	Chicago	IL	60608
1208 N Wells ST	Chicago	IL	60610

	DBAName	AKAName	Address	City	State	Zip
t1	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	60608
t2	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	**60608**
t3	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	**60608**
t4	**Johnnyo's**	Johnnyo's	3465 S Morgan ST	**Chicago**	IL	60608

External dictionaries may have limited coverage or not exist altogether

Quantitative Statistics

Reason about co-occurrence of values across cells in a tuple

Estimate the distribution governing each attribute

	DBAName	AKAName	Address	City	State	Zip
t1	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	***Chicago***	IL	*60608*
t2	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	***60609***
t3	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	***60609***
t4	***Johnnyo's***	Johnnyo's	3465 S Morgan ST	***Cicago***	IL	60608

Example: Chicago co-occurs with IL

Hellerstein, 2008; Mayfield et al., 2010; Yakout et al., 2013

Quantitative Statistics

Reason about co-occurrence of values across cells in a tuple

Estimate the distribution governing each attribute

	DBAName	AKAName	Address	City	State	Zip
t1	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	60608
t2	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	60609
t3	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	60609
t4	**John Veliotis Sr.**	Johnnyo's	3465 S Morgan ST	**Chicago**	IL	60608

Again, fails to repair the wrong zip code

Combining Everything

Constraints and minimality

	DBAName	AKAName	Address	City	State	Zip
t1	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	**60609**
t2	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	60609
t3	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	60609
t4	Johnnyo's	Johnnyo's	3465 S Morgan ST	Cicago	IL	60608

External data

	DBAName	AKAName	Address	City	State	Zip
t1	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	60608
t2	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	**60608**
t3	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	**60608**
t4	Johnnyo's	Johnnyo's	3465 S Morgan ST	**Chicago**	IL	60608

Quantitative statistics

	DBAName	AKAName	Address	City	State	Zip
t1	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	60608
t2	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	60609
t3	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	60609
t4	**John Veliotis Sr.**	Johnnyo's	3465 S Morgan ST	**Chicago**	IL	60608

Different solutions suggest different repairs

HoloClean: a probabilistic model for data repairs

	Address	City	State	Zip
t1	3465 S Morgan ST	Chicago	IL	60608
t2	3465 S Morgan ST	Chicago	IL	60609
t3	3465 S Morgan ST	Chicago	IL	60609
t4	3465 S Morgan ST	Cicago	IL	60608

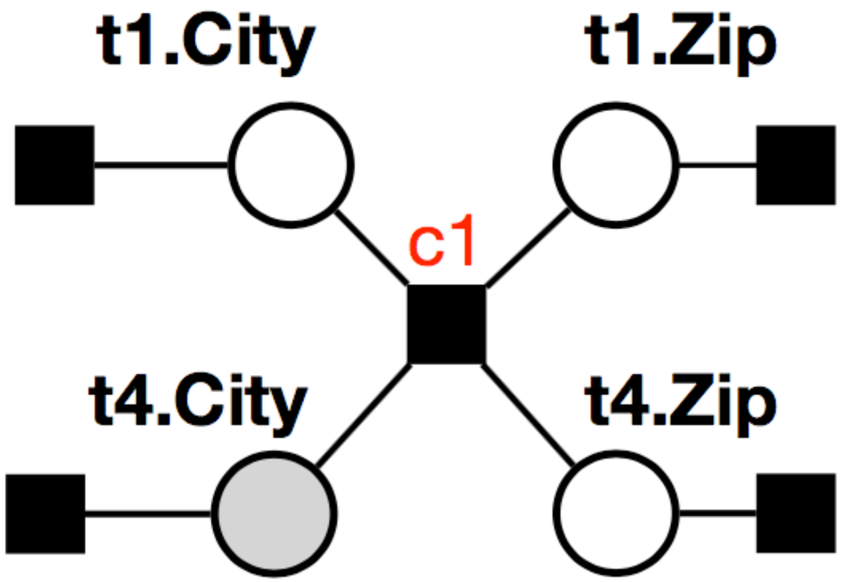
Each cell is a random variable

Value co-occurences capture data statistics

Constraints introduce correlations

c1: Zip $\rightarrow$ City

“Address= 3465 S Morgan St”



: Unknown (to be inferred) RV

: Observed (fixed) RV

: Factor (encodes correlations)

[HoloClean: Holistic Data Repairs with Probabilistic Inference](). [VLDB'17]

Input

Dataset to be cleaned

	DBAName	Address	City	State	Zip
t1	John Veliotis Sr.	3465 S Morgan ST	Chicago	IL	60608
t2	John Veliotis Sr.	3465 S Morgan ST	Chicago	IL	60609
t3	John Veliotis Sr.	3465 S Morgan ST	Chicago	IL	60609
t4	Johnnyo's	3465 S Morgan ST	Cicago	IL	60608

Denial Constraints

c1: DBAName $\rightarrow$ Zip
c2: Zip $\rightarrow$ City, State
c3: City, State, Address $\rightarrow$ Zip

Matching Dependencies

m1: $Zip = Ext\_Zip \rightarrow City = Ext\_City$
m2: $Zip = Ext\_Zip \rightarrow State = Ext\_State$
m3: $City = Ext\_City \wedge State = Ext\_State \wedge$
$\wedge\ Address = Ext\_Address \rightarrow Zip = Ext\_Zip$

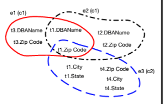
External Information

Ext_Address	Ext_City	Ext_State	Ext_Zip
3465 S Morgan ST	Chicago	IL	60608
1208 N Wells ST	Chicago	IL	60610
259 E Erie ST	Chicago	IL	60611
2806 W Cermak Rd	Chicago	IL	60623

The HoloClean Framework

1. Error Detection Module

- Use integrity constraints
- Leverage external data
- Detect outliers
- Identify possible repairs

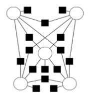


2. Compilation Module

- Automatic Featurization
- Statistical analysis and candidate repair generation
- Compilation to probablistic program

3. Repair Module

- Ground probalistic model
- Statistical learning
- Probabilistic inference



Output

Proposed Cleaned Dataset

	DBAName	Address	City	State	Zip
t1	John Veliotis Sr.	3465 S Morgan ST	Chicago	IL	60608
t2	John Veliotis Sr.	3465 S Morgan ST	Chicago	IL	**60608**
t3	John Veliotis Sr.	3465 S Morgan ST	Chicago	IL	**60608**
t4	**John Veliotis Sr.**	3465 S Morgan ST	**Chicago**	IL	60608

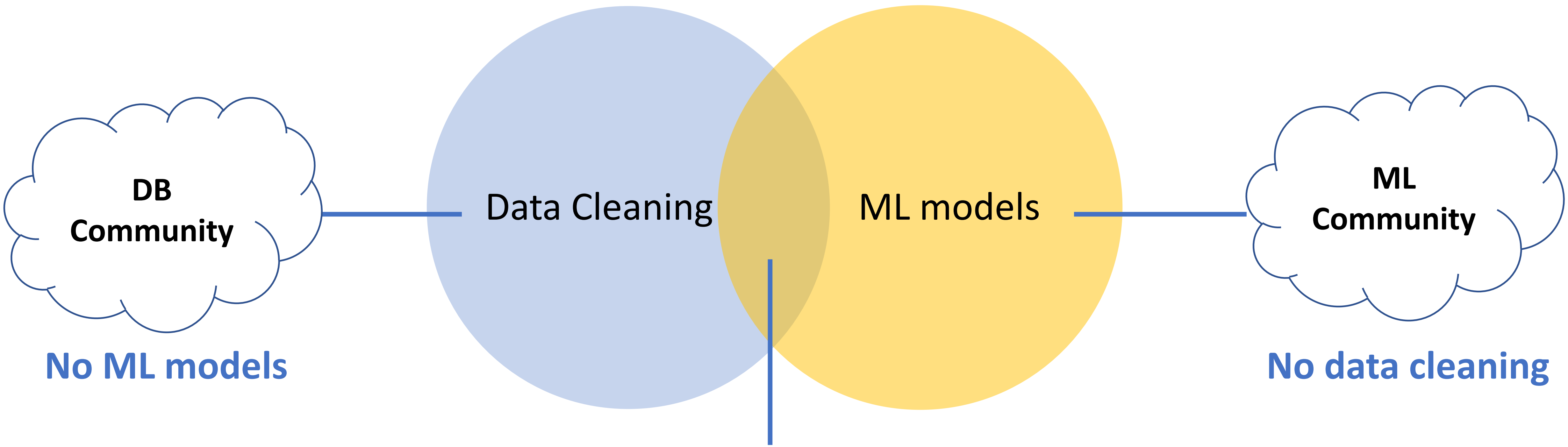
Marginal Distribution of Cell Assignments

Cell	Possible Values	Probability
t2.Zip	60608	0.84
	60609	0.16
t4.City	Chicago	0.95
	Cicago	0.05
t4.DBAName	John Veliotis Sr.	0.99
	Johnnyo's	0.01

Probabilistic model that unifies different signals for repairing a dataset.

Adapted from Intro to Data Cleaning lecture from Prof. Xu Chu

Data cleaning and ML



The impact of data cleaning on downstream ML models?

Data cleaning and ML

Cleaning “for” ML:

- Leverage the downstream ML model or application to define cleaning signals that incorporates high-level semantics
- Why is this a good idea?
  - Clean datasets that contain fully correct attributes are rarely available
  - Data cleaning can sometimes negatively impact the performance of ML models
    - CleanML: A Study for Evaluating the Impact of Data Cleaning on ML Classification Tasks
- Example: BoostClean: Automated Error Detection and Repair for Machine Learning

Reading: From Cleaning Before ML to Cleaning For ML

Data Labeling

Data is the Bottleneck for ML

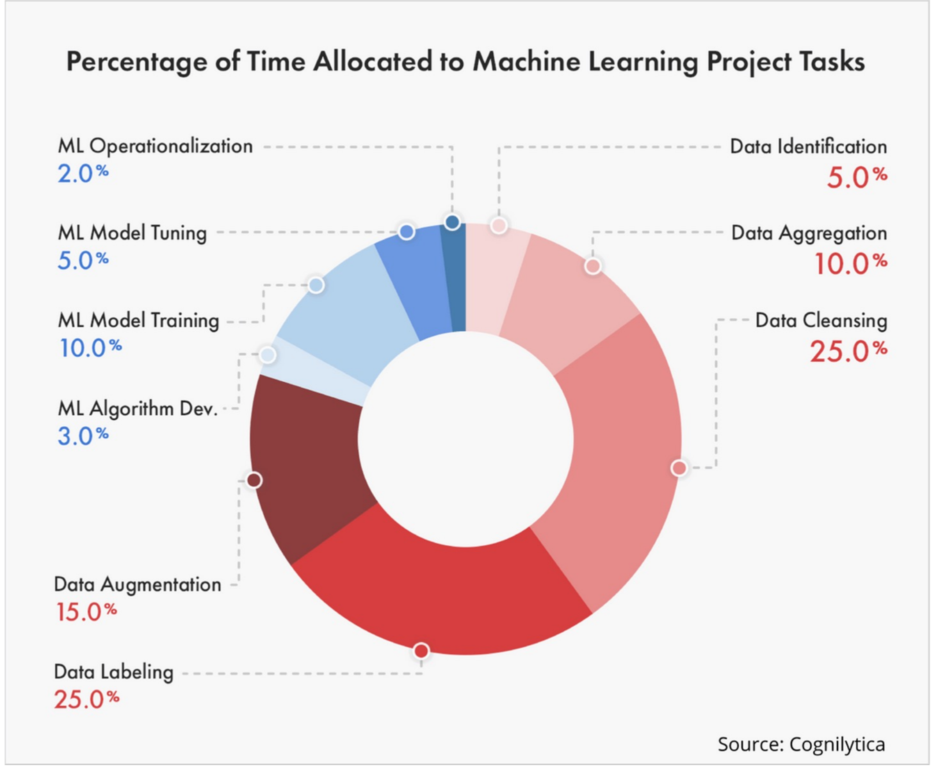
ML ≈ Model + Data

Model is gradually commoditized

- Out-of-the-box invocation of ML libraries gives decent results
- Transformers for "all" tasks

Data is the bottleneck

OpenAI has hired an army of contractors
to do what's called "data labeling"



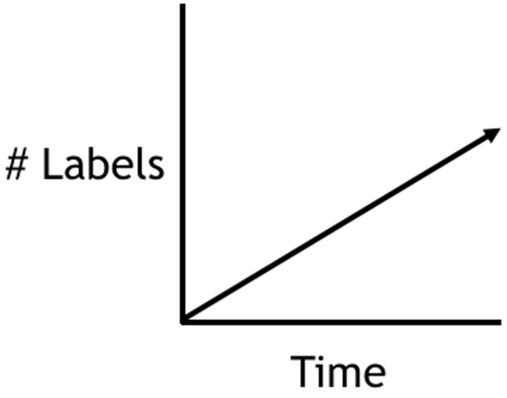
Sources:
https://www.semafor.com/article/01/27/2023/openai-has-hired-an-army-of-contractors-to-make-basic-coding-obsolete
https://www.datanami.com/2023/01/20/openai-outsourced-data-labeling-to-kenyan-workers-earning-less-than-2-per-hour-time-report/

Manual v.s. Programmatic Labeling

Labeling individual data points

Writing Labeling Functions (LFs) where each LF abstracts a supervision source (e.g. heuristics, existing models, external KBs, …)

```
@labeling_function()
def lf_contains_link(x):
    # Return a label of SPAM if "http" in comment text, otherwise ABSTAIN
    return SPAM if "http" in x.text.lower() else ABSTAIN
```

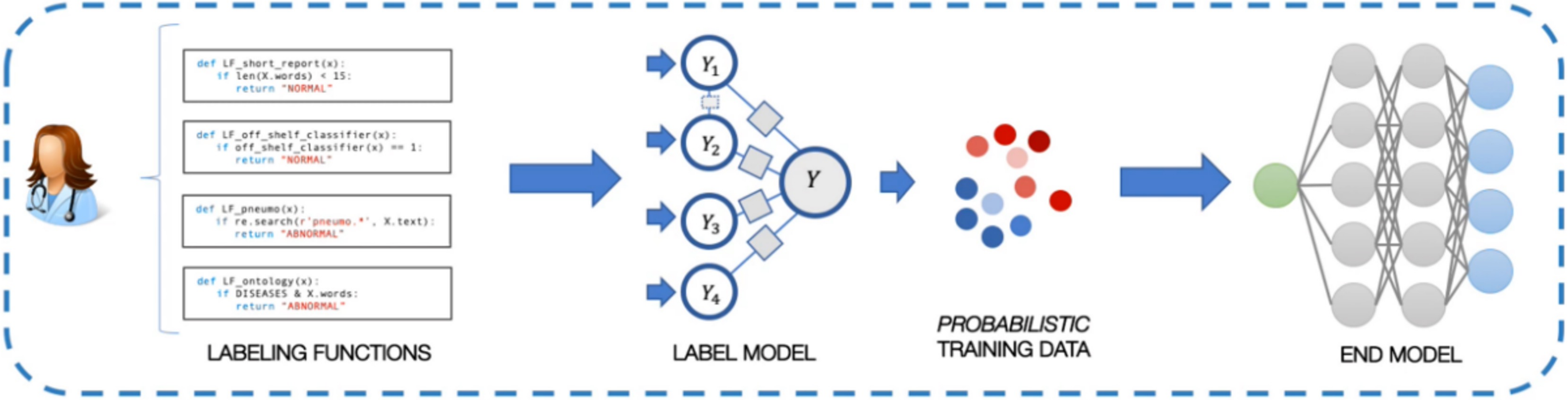


Exact Labels

Source: https://www.snorkel.org/use-cases/01-spam-tutorial#3-writing-more-labeling-functions

Programmatic Labeling Pipeline Overview

Credit: Snorkel Project



(1) Users **write labeling functions** to generate noisy labels

(2) A **label model combines noisy labels** to be probabilistic labels

(3) Using the **probabilistic labels to train** an end ML model

(1) Labeling Function

"Indication: Chest pain. Findings: Focal consolidation and pneumothorax.."

```
def LF_pneumothorax(c):
  if re.search(r'pneumo.*', c.report.text):
     return "ABNORMAL"
```

"Indication: Chest pain. Findings: No focal consolidation or pneumothorax..."

LFs can be noisy!

Source: https://ajratner.github.io/assets/papers/Snorkel_VLDB_2018_slides.pdf

Other Example LFs: Existing Knowledge

- Knowledge bases
  - Match the text inputs against the knowledge base (e.g., DBPedia) to search for known spouse relationships.
- Pretrained models
  - Pre-trained model with a different label space
- Thirty-party tools
  - TextBlob: Simplified Text Processing

How are LFs developed

- By domain experts
- Generate programmatically
  - Snuba: Automating Weak Supervision to Label Training Data. [VLDB'18]
  - Language Models Enable Simple Systems for Generating Structured Views of Heterogeneous Data Lakes

(2) Label Model

1: positive
-1: negative
0: abstain

	LF1	LF2	LF3	LFX
Data point 1	1	-1	0	…
Data point 2	0	1	0	…
Data point 3	1	1	-1	..
Data point 4	1	1	-1	…
Data point 5	1	1	-1	…
Data point x	…	…	…	…

Weak label matrix X

$\xrightarrow{\text{Label model}}$

y
1
1
-1
1
-1
…

Inferred ground-truth labels y

Example label model

Option 1: Majority voting

Q: What if some rules are more reliable than others?

Option 2: Evaluate the accuracy of each labeling function

Example: Dawid and Skene's method

1. Assume accuracies θ of each LF
2. Learn parameter θ with an Expectation and Maximization algorithm:
   a. Initialize y by majority vote
   b. Calculate accuracies θ for each LF
   c. Update y by maximizing $p(X|y, \theta)$