Pyramid

Enhancing Selectivity in Big Data Protection

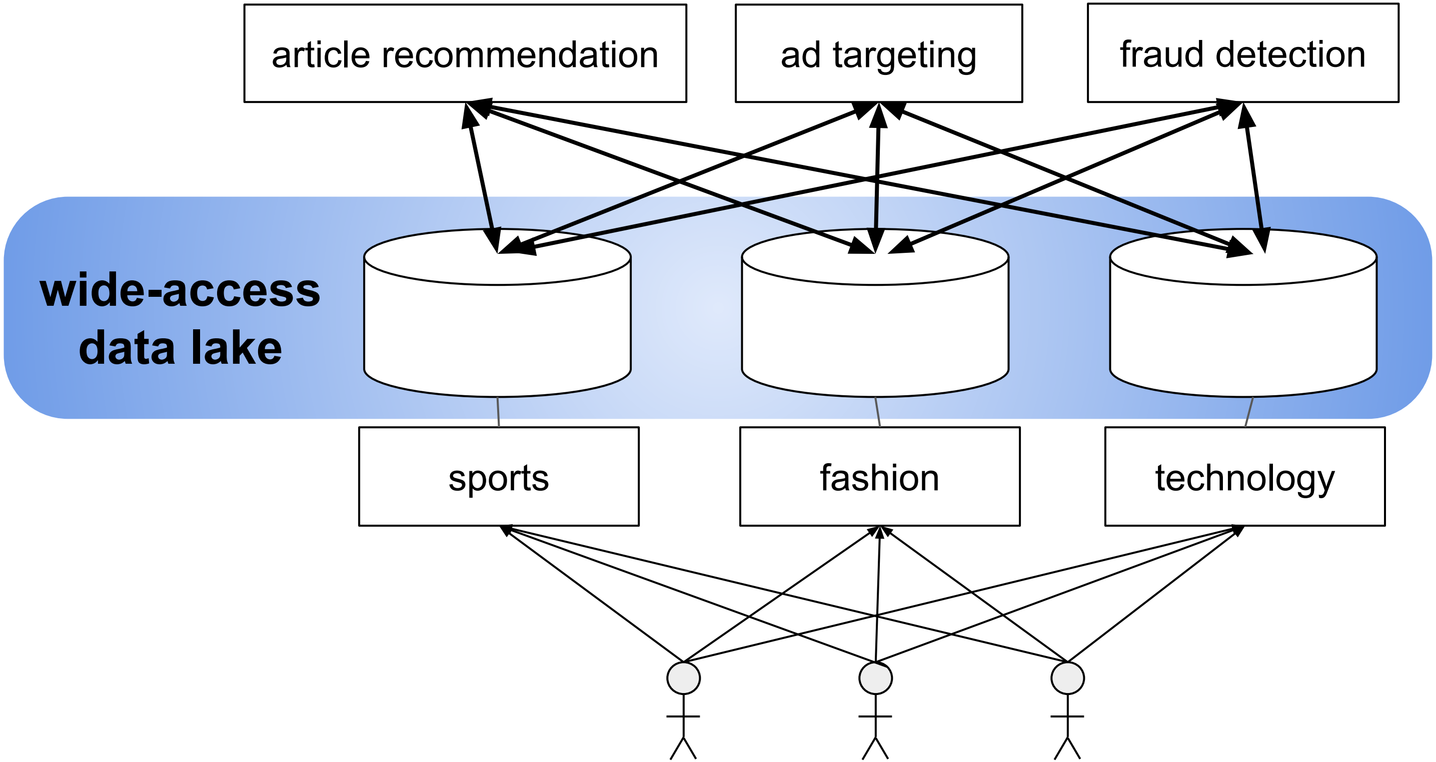
Mathias Lécuyer, **Riley Spahn**,
Roxana Geambasu, Tzu-Kuo Huang, Siddhartha Sen

Columbia University

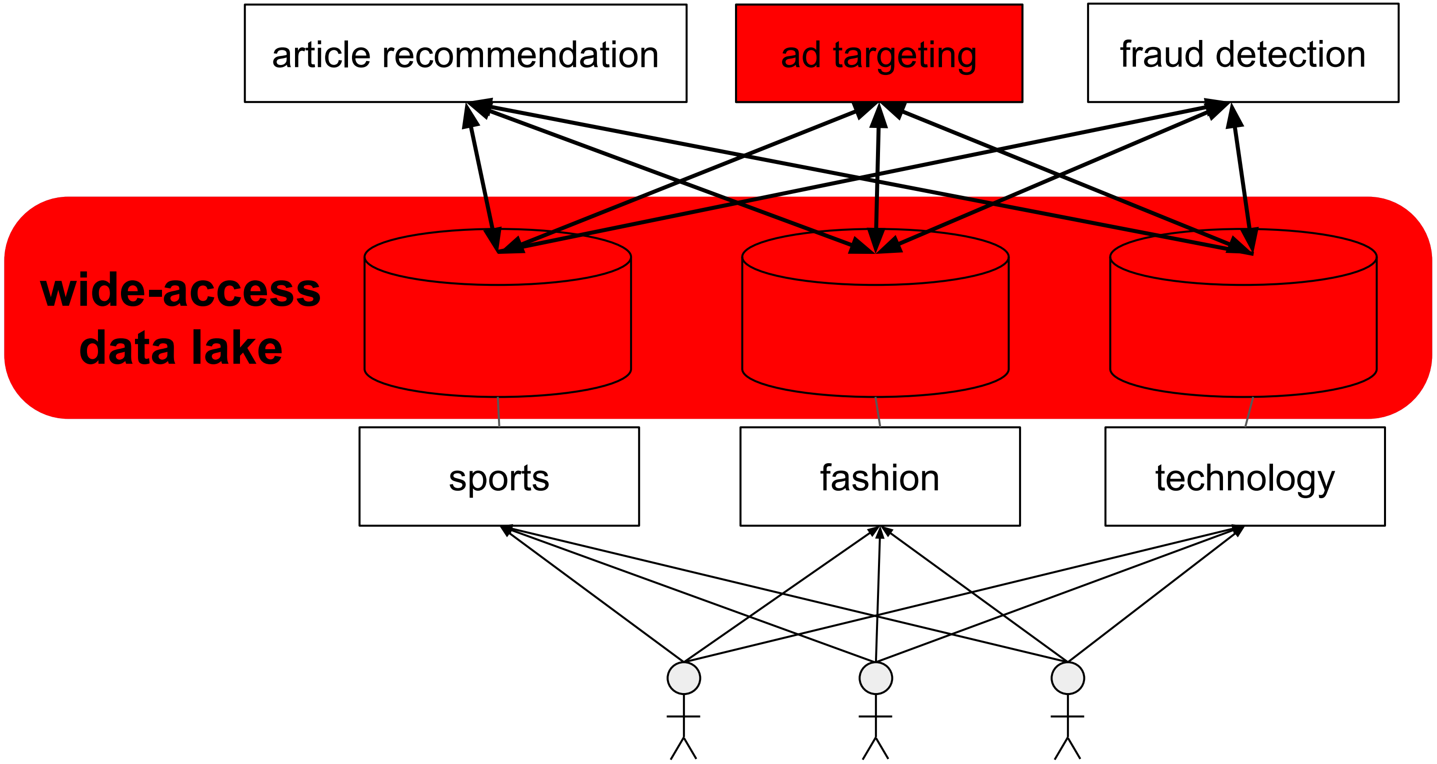
The “Collect-Everything” Mentality

- Companies collect enormous personal data
  - Clicks, location, browsing history, many more
- Data has beneficial uses
  - Article recommendation
  - Ad targeting
  - Fraud detection
- But data raises substantial risks in the event of a breach

The "Data Lake" Mentality



Collection + Wide Access Lead to Exposure



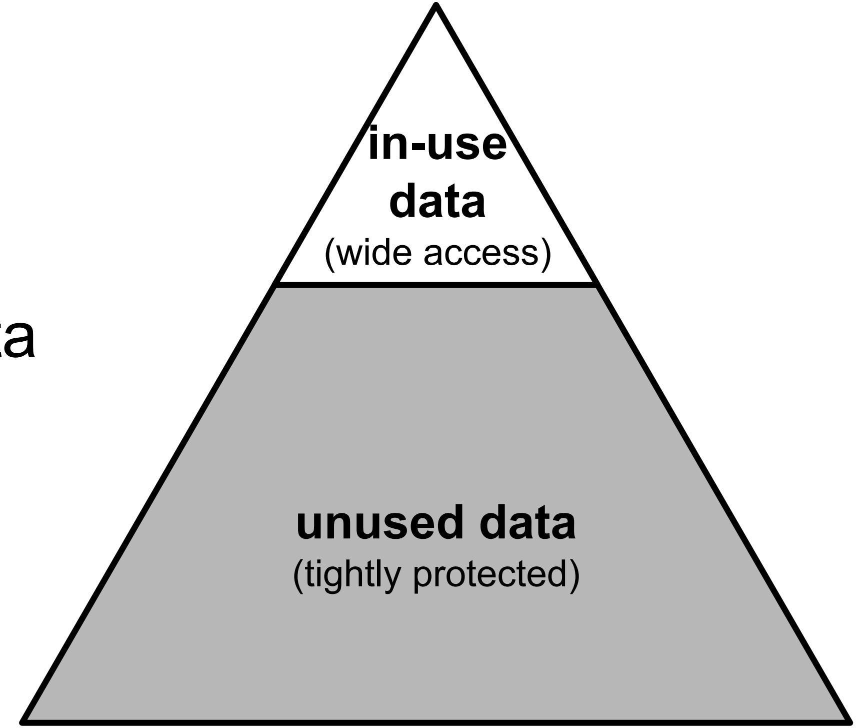
Question: Can Companies Be More Selective?

- We hypothesize that not all data that is collected is needed or used.

- If we can distinguish “needed” data from “unneeded” data, we can greatly improve protection.
  - E.g., store unneeded data offline

Selective Data Systems

1. Limit in-use data
2. Avoid accessing unused data
3. Without impacting accuracy, performance



How to achieve selectivity in machine learning?

- Access to the “working set” is not enough
- (Re)training models requires access to most/all data

- Training set minimization addresses this
  - E.g.: sampling, count featurization, active learning, ...
  - Can we retrofit these mechanisms for protection?

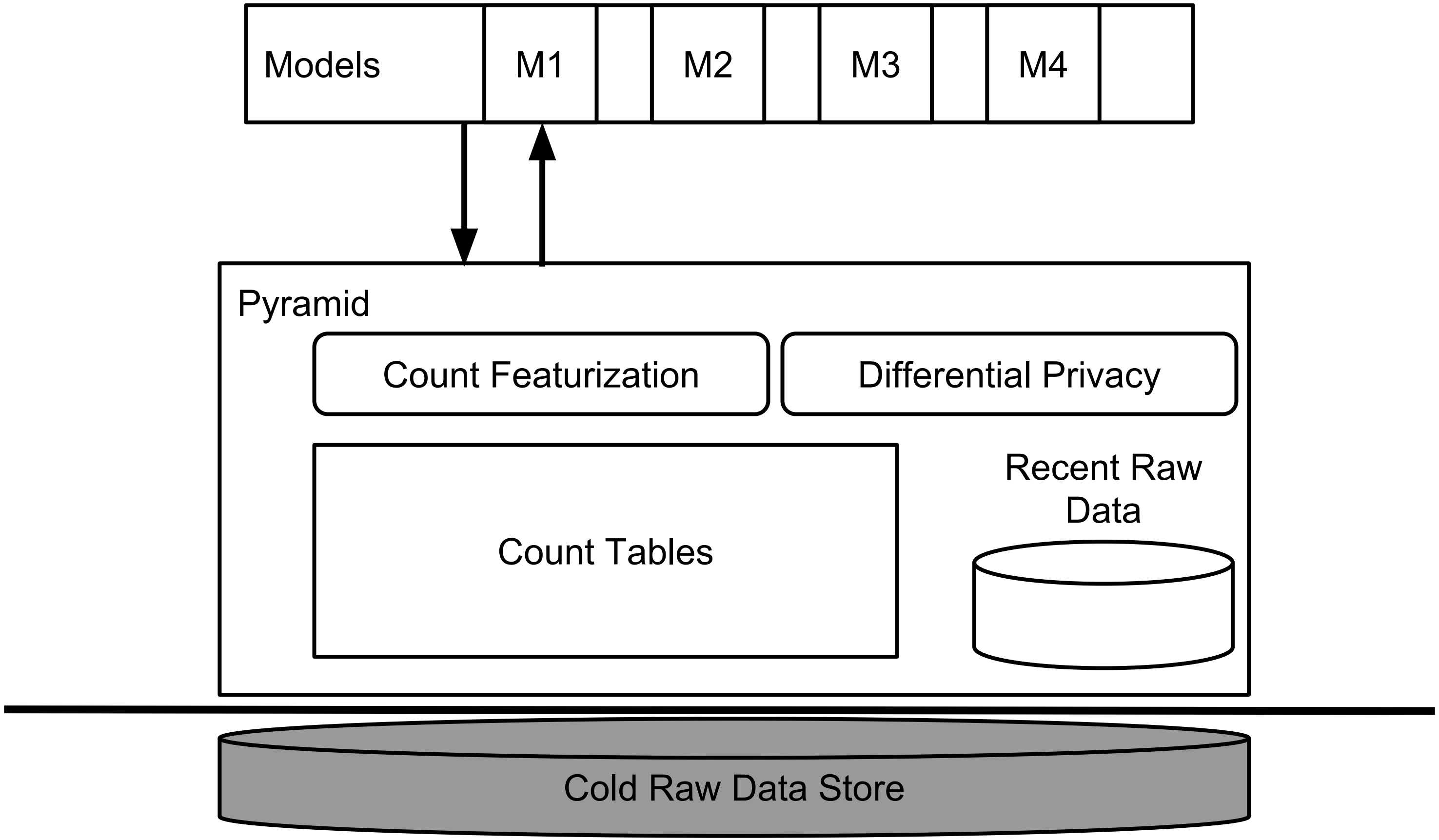
Pyramid

- First selective data system
- Retrofits count featurization for protection
  - Keeps a small amount of recent raw data
  - Summarizes past data using differentially private count tables
  - Combines the raw data with count features and feeds that into ML models for training
- Reduces data exposure by two orders of magnitude with moderate performance degradation

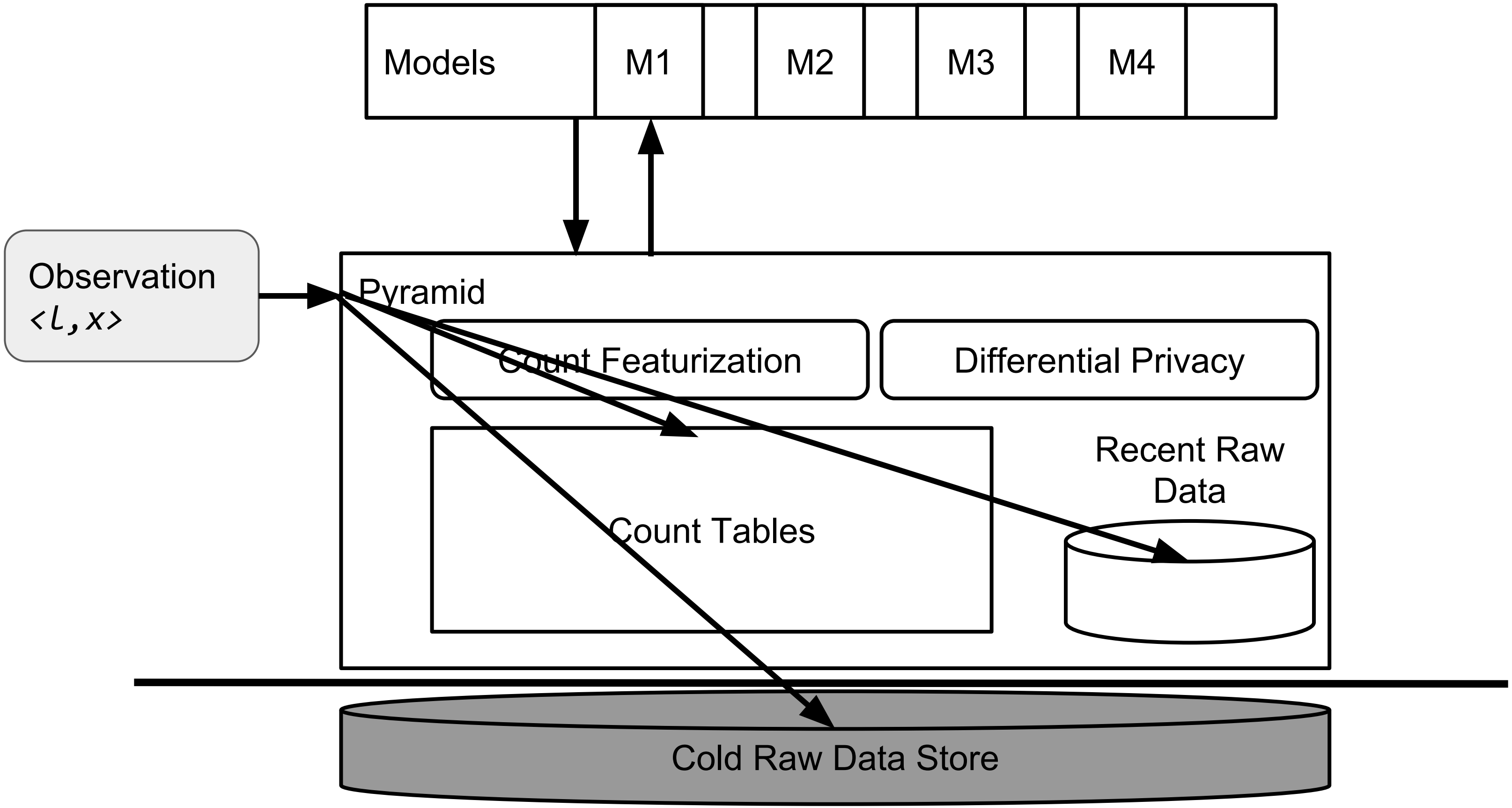
Outline

- Motivation
- **Design**
- Evaluation
- Conclusions

Architecture



Architecture



Count Featurization Example

AdId		
Value	Click	No Click
A1	0	0
A2	0	0

UserID		
Value	Click	No Click
U1	0	0
U2	0	0

PageID		
Value	Click	No Click
P1	0	0
P2	0	0

Count Featurization Example

AdId		
Value	Click	No Click
A1	1	0
A2	0	0

UserID		
Value	Click	No Click
U1	1	0
U2	0	0

PageID		
Value	Click	No Click
P1	1	0
P2	0	0

<Label:Click | AdId:A1, UserID:U1, PageID:P1>

Count Featurization Example

AdId		
Value	Click	No Click
A1	1	0
A2	0	1

UserID		
Value	Click	No Click
U1	1	1
U2	0	0

PageID		
Value	Click	No Click
P1	1	0
P2	0	1

<Label:Click | AdId:A1, UserID:U1, PageID:P1>

<Label:No-Click | AdId:A2, UserID:U1, PageID:P2>

Count Featurization Example

AdId		
Value	Click	No Click
A1	1	1
A2	0	1

UserID		
Value	Click	No Click
U1	1	1
U2	0	1

PageID		
Value	Click	No Click
P1	1	0
P2	0	2

<Label:Click | AdId:A1, UserID:U1, PageID:P1>

<Label:No-Click | AdId:A2, UserID:U1, PageID:P2>

<Label:No-Click | AdId:A1, UserID:U2, PageID:P2>

Count Featurization Example

AdId		
Value	Click	No Click
A1	1250	23751
A2	1482	26765

UserID		
Value	Click	No Click
U1	105	1523
U2	112	1288

PageID		
Value	Click	No Click
P1	1300	63700
P2	3692	29874

Count Featurization Example

AdId		
Value	Click	No Click
A1	1250	23751
A2	1482	26765

UserID		
Value	Click	No Click
U1	105	1523
U2	112	1288

PageID		
Value	Click	No Click
P1	1300	63700
P2	3692	29874

<AdId:A1, UserID:U2, PageID:P1>

↓

<P(click|AdId=A1), P(click|UserID=U2), P(click|PageID=P1)>

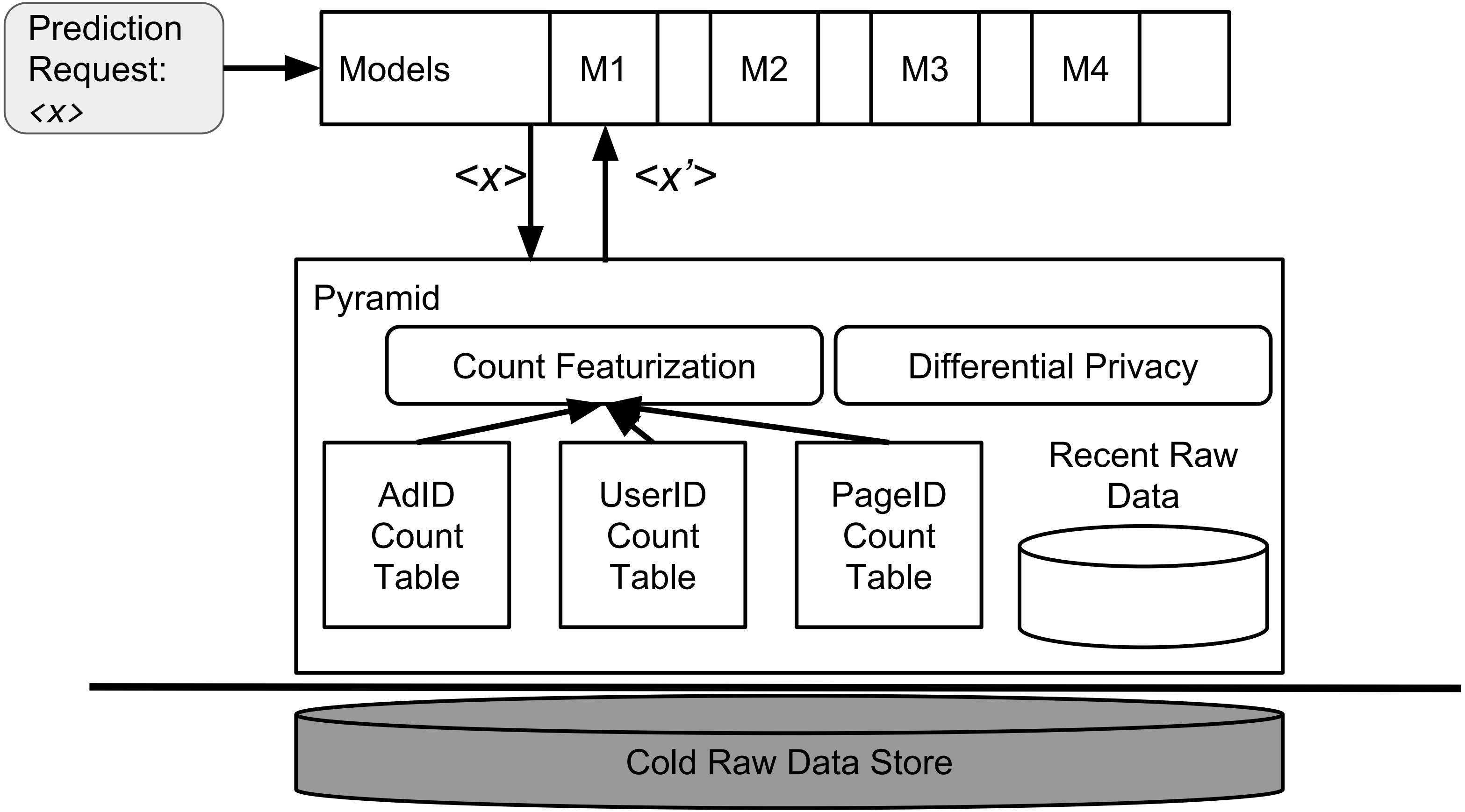
↓

<0.05, 0.08, 0.02>

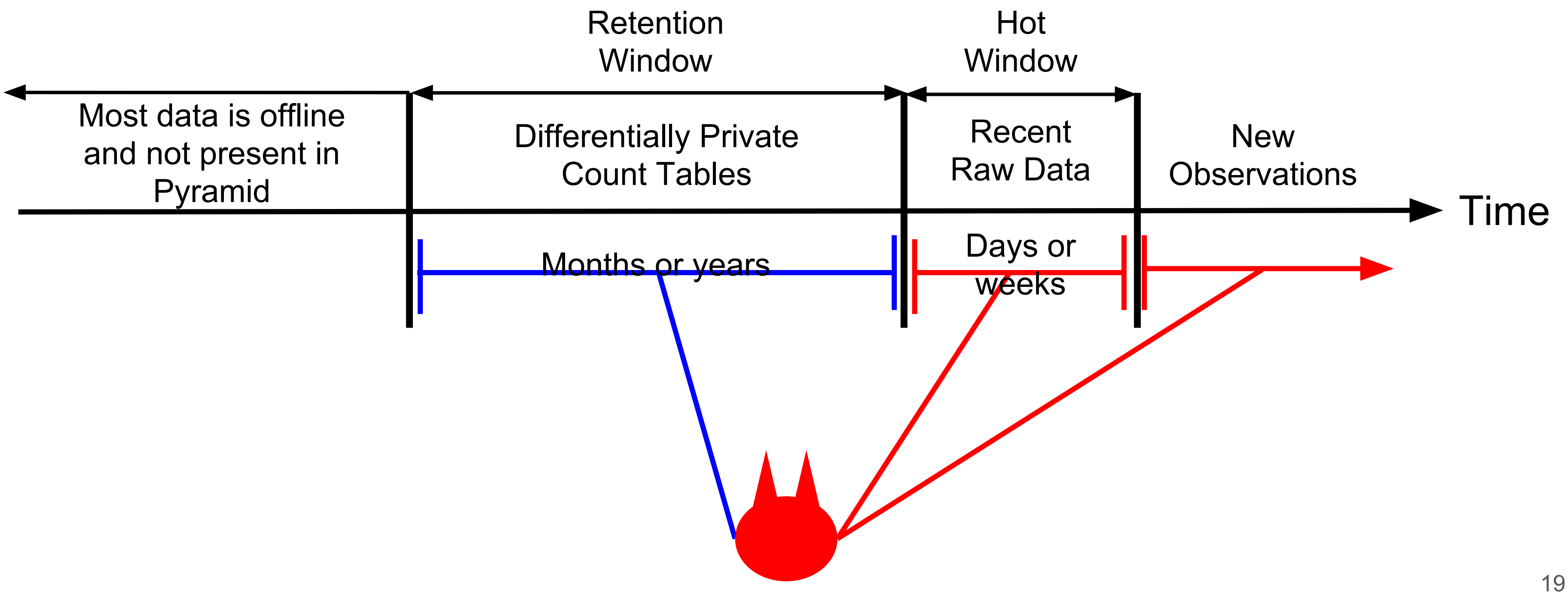
↓

ML Model

Architecture



Pyramid's Protections



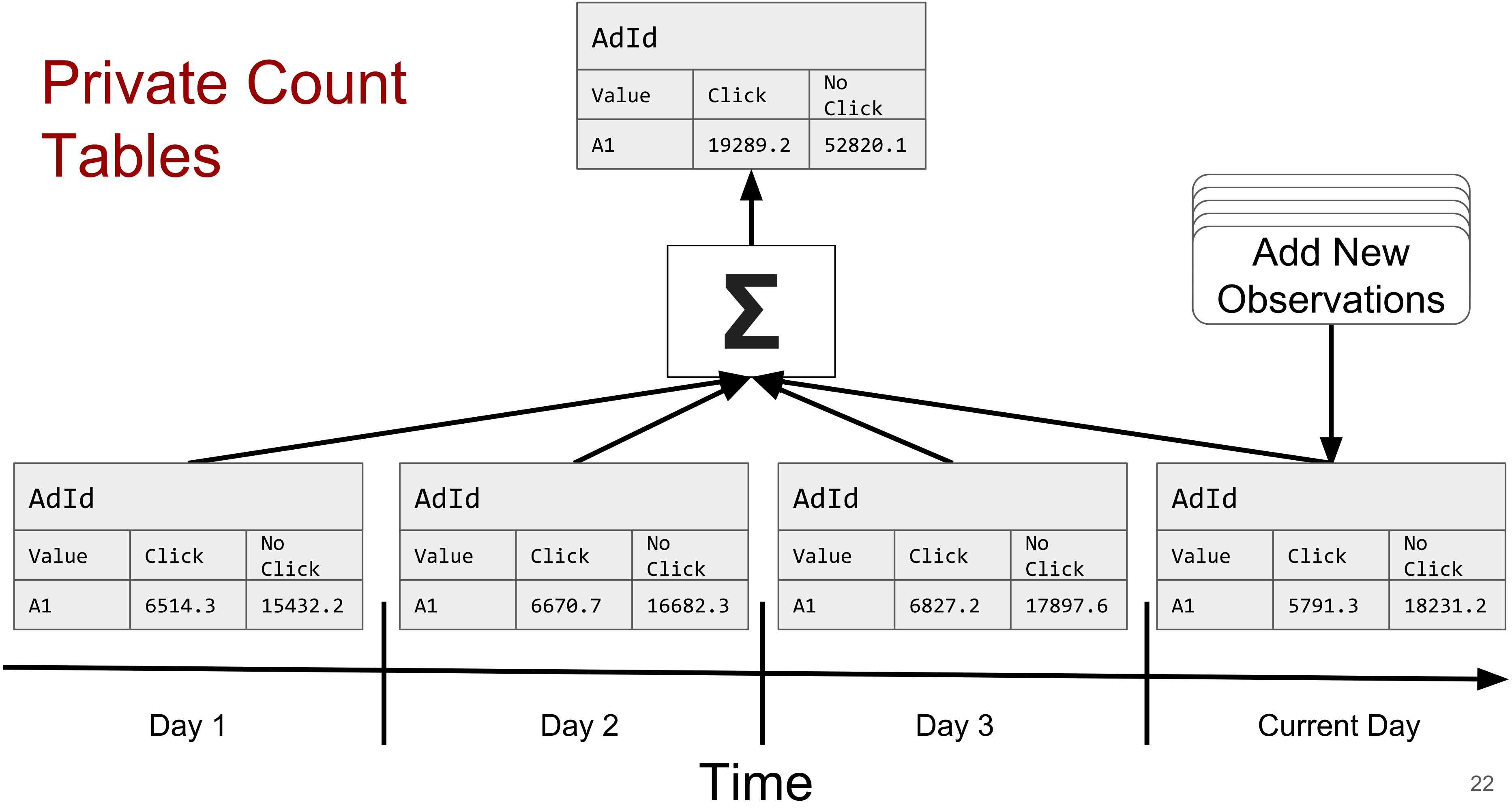
Protection Assumptions

- State is not managed out of band
- Models are retrained on request
- State from previous models does not persist

Differential Privacy

- Randomizes output to protect privacy
- Privacy budget, ε, shared among queries
- Resilient to auxiliary information
- Resilient to post-processing

Private Count Tables



Challenges Combing Count Featurization and Differential Privacy

Challenge	Solution
● Support large datasets with large numbers of features	● Private Count-Median Sketch
● Must choose optimal count tables to support future workloads	● Feature Combination Selection
● Some features are more sensitive to differential privacy	● Weighted Noise Infusion

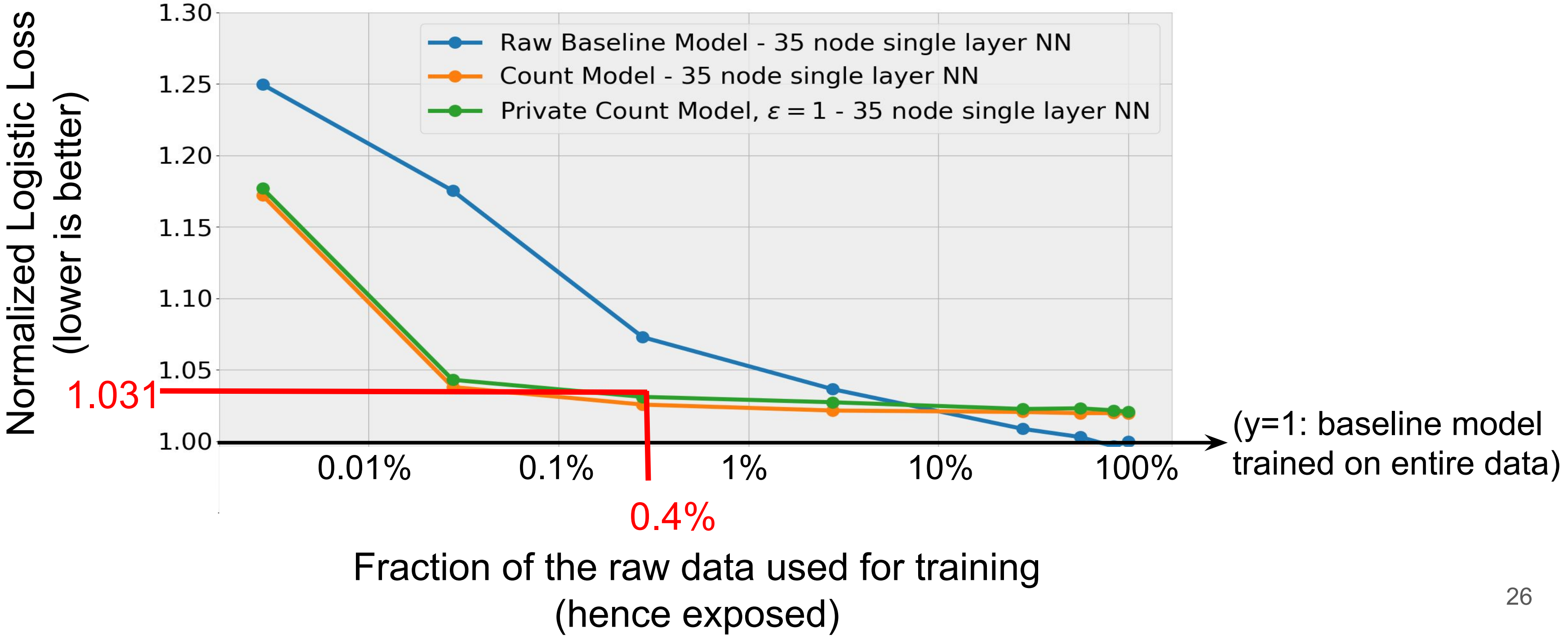
Outline

- Motivation
- Design
- **Evaluation**
- Conclusions

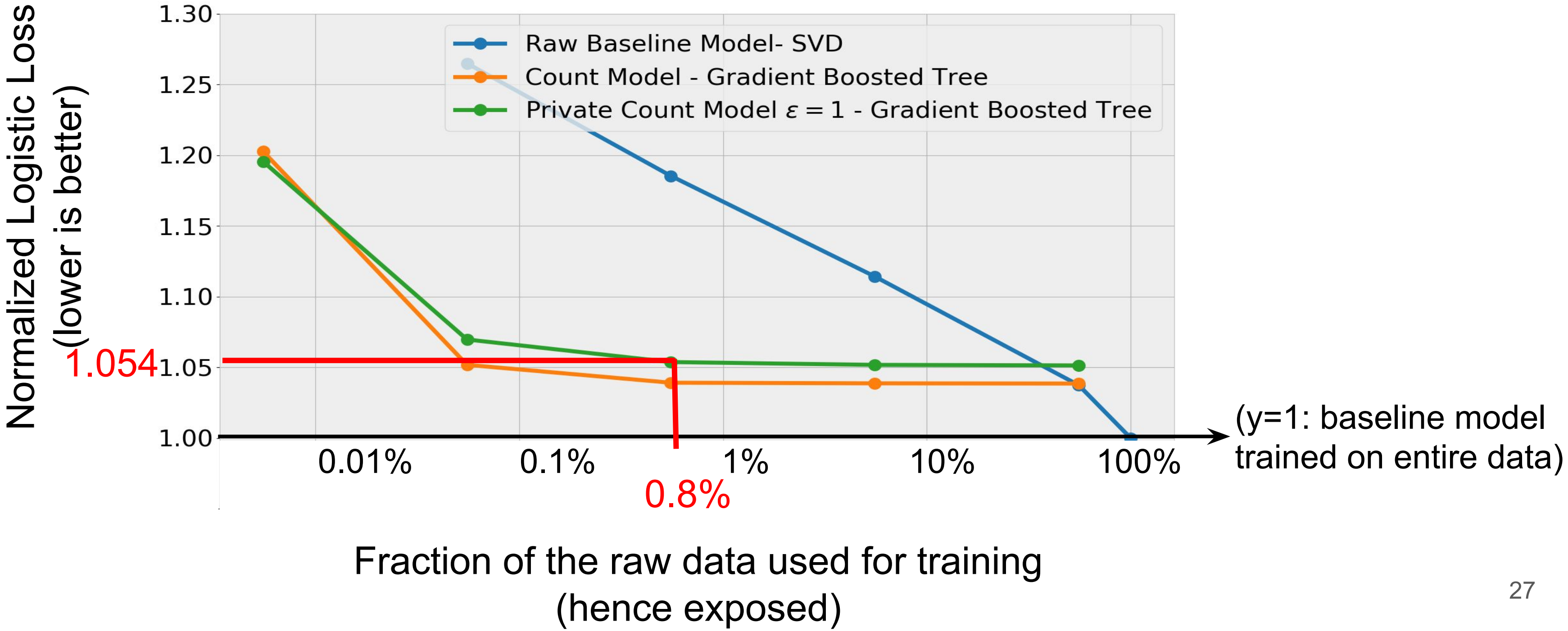
Evaluation Datasets

- Criteo
  - Ad click/no-click prediction
  - Estimating probability of a click
  - 45 million points w/ 39 features
- Movielens
  - Movie rating prediction
  - Estimate probability a user will rate a movie highly
  - 22 million ratings, 34K movies, 240K users

Criteo: Training on just 0.4% of the data leads to only 3.1% loss in accuracy

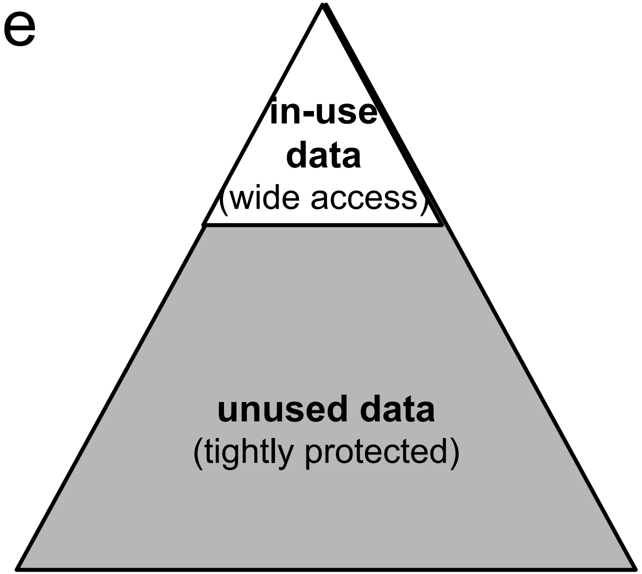


Movielens: Training on just 0.8% of the data leads to only 5.4% loss in accuracy



Conclusions

- Data collection and wide access increase exposure risks
- Selective data systems minimize in-use data and separate it from unused data
  - Training set minimization is a productive way to think about selectivity
- Pyramid retrofits count featurization for protection with differential privacy
  - Reduces exposure 2 orders of magnitude



Limitations and Future Work

- Pyramid applicability:
  - Works well for classification problems
  - Most effective for categorical features
  - Supports some but not all workload evolutions
- Future: extend applicability by retrofitting other training set minimization mechanisms for protection
  - Vector quantization: can support continuous features
  - Sampling and herding: can support unsupervised tasks
  - Active learning: can permit selective data collection