FEATPILOT: Automatic Feature Augmentation on Tabular Data

Jiaming Liang

University of Pennsylvania

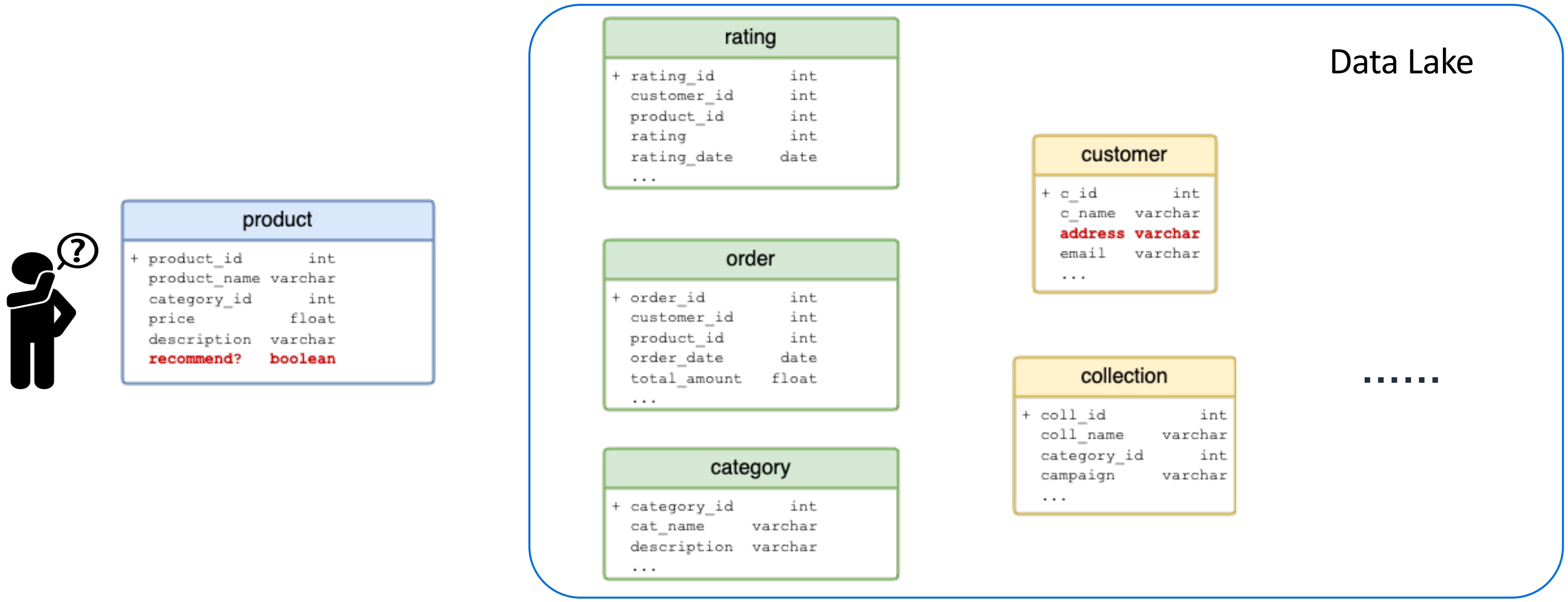
liangjm@seas.upenn.edu

Chuan Lei, Xiao Qin, Jiani Zhang, Asterios Katsifodimos, Christos Faloutsos, Huzefa Rangwala

Amazon Web Services

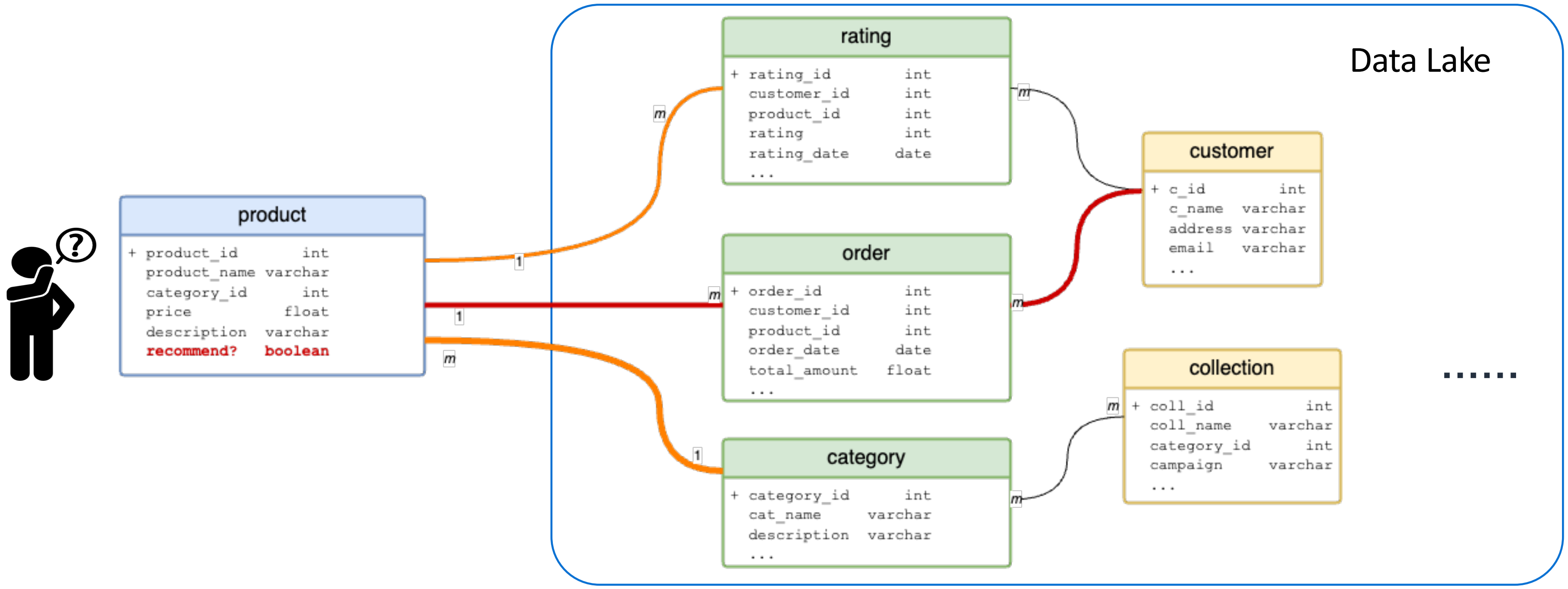
chuanlei, drxqin, zhajiani, akatsifo, faloutso, rhuzefa@amazon.com

Motivation



- Informatics-driven decision making / data-centric ML
- Useful features live in massive enterprise/open data lake

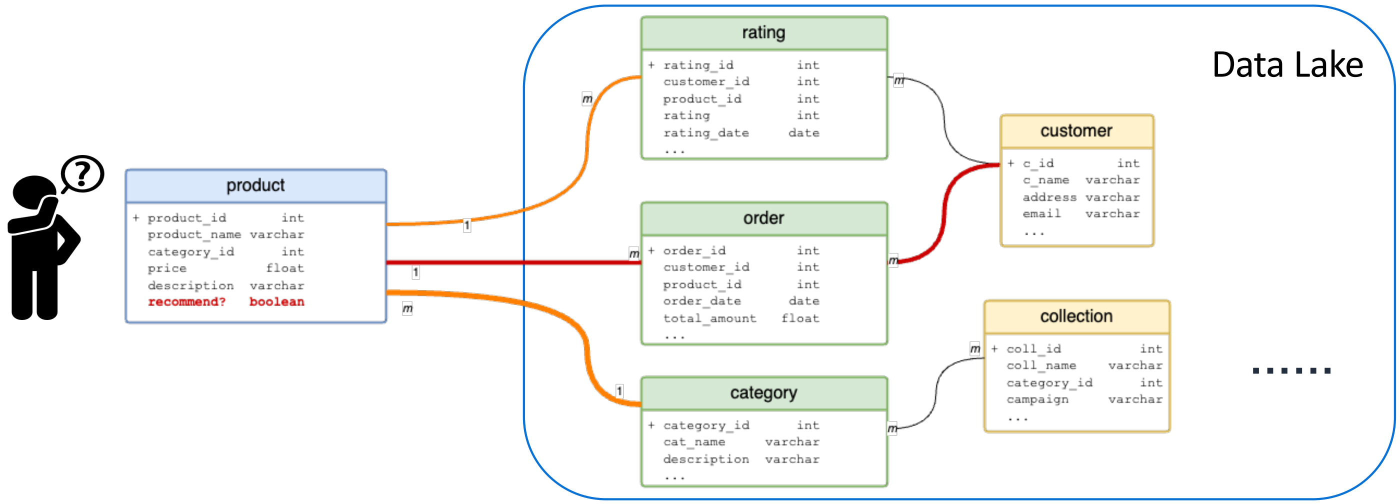
Motivation



- Informatics-driven decision making / data-centric ML.
- Useful features live in massive enterprise/open data lake.

Automatic Feature Augmentation!

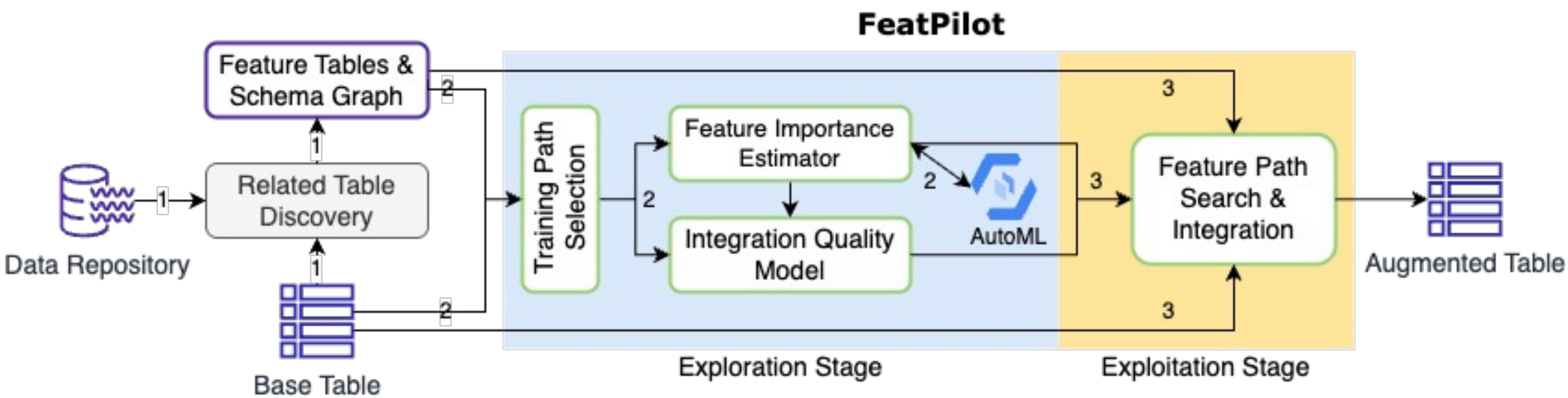
Challenges & Limitations



- Massive data
- Complex join relationship
- Massive feature combination
- ML task complexity

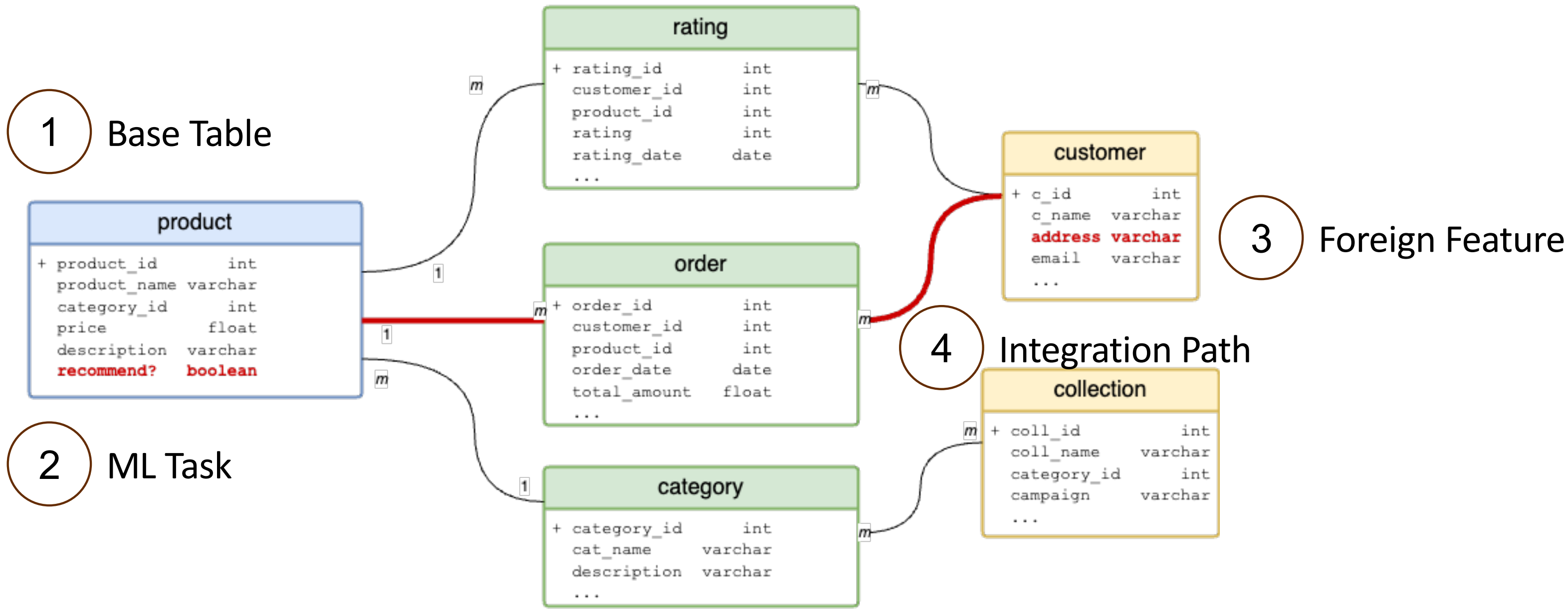
Massive search space!

Contributions



- A decomposition method for intractable feature augmentation tasks
- FeatPilot, an automatic feature augmentation system
- 10.27% ML performance over SOTA methods on six public datasets

Problem Definition

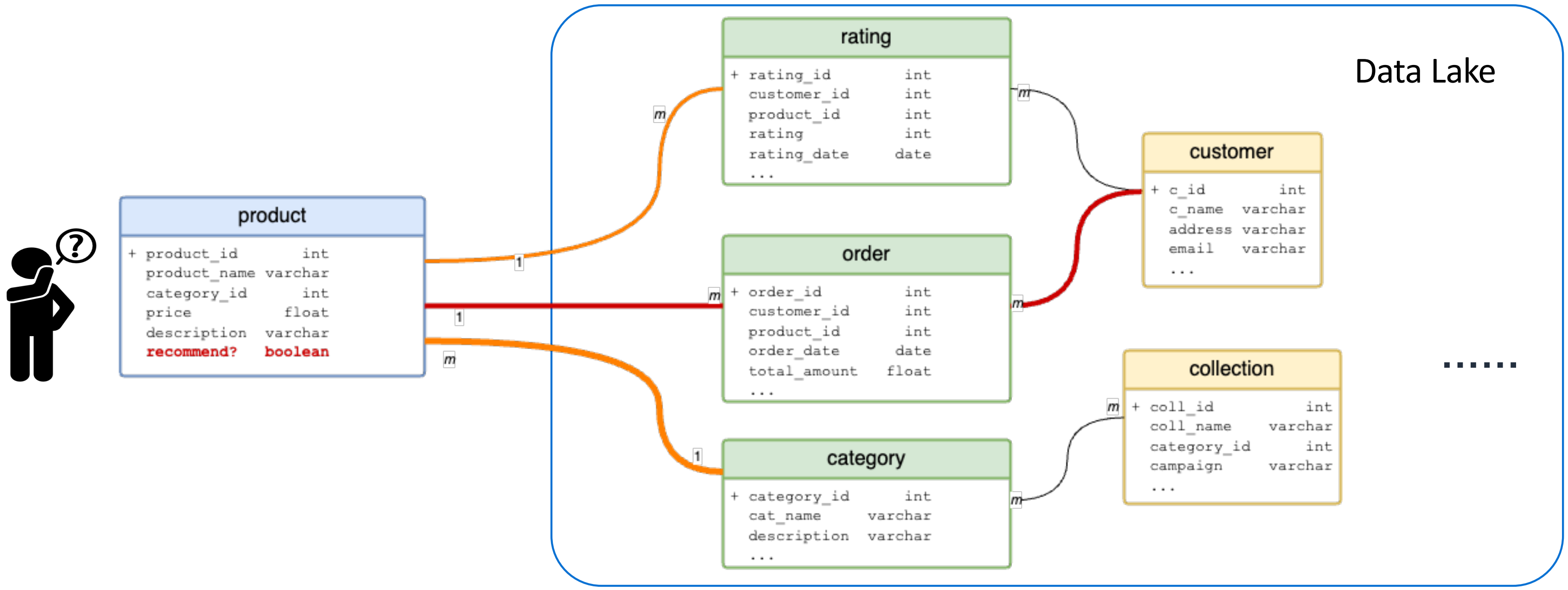


5 Utility Gain = Score(product ⋈ order ⋈ customer) - Score(product)

Objective: Maximize Utility Gain with an integration strategy

$$\max_{\mathcal{S}\subseteq\mathcal{T},\mathcal{P}} UG = US(T_{aug}) - US(T_{base})$$
$$\text{subject to } T_{aug} = T_{base} \bowtie_{\mathcal{P}} \mathcal{S}$$

Key Idea: Accessing a feature's value



One feature's value
- How many instances can get this feature? (Integration Quality: IQ)
- The relationship between the feature and ML task. (Feature Importance: FI)

Key Idea: Integration Quality definition

product

product_id	name	recommend
p1	iPhone 15	?
p2	Pixel 8	?
p3	Galaxy S23	?

transaction_1

transaction_id	product_id	amount
t1	p1	799.99
t2	p2	699.99

transaction_2

transaction_id	product_id	amount
t3	p3	999.99

rating

transaction_id	rating_id	rating
t1	r1	5
t2	r2	4.5
t3	r3	4
t4	r4	5

Join Graph

	product_id	name	rating	recommend
non-NA	p1	iPhone 15	5	?
non-NA	p2	Pixel 8	4.5	?
NA	p3	Galaxy S23	null	?

T_{aug1}

	product_id	name	rating	recommend
NA	p1	iPhone 15	null	?
NA	p2	Pixel 8	null	?
non-NA	p3	Galaxy S23	4	?

T_{aug2}

Integration Quality: Percentage of instances getting the target feature

E.g. $IQ(T_{aug1}) = \frac{2}{3}$

Key Idea: Integration Quality definition

product

product_id	name	recommend
p1	iPhone 15	?
p2	Pixel 8	?
p3	Galaxy S23	?

transaction_1

transaction_id	product_id	amount
t1	p1	799.99
t2	p2	699.99

transaction_2

transaction_id	product_id	amount
t3	p3	999.99

rating

transaction_id	rating_id	rating
t1	r1	5
t2	r2	4.5
t3	r3	4
t4	r4	5

Join Graph

product_id	name	rating	recommend
p1	iPhone 15	5	?
p2	Pixel 8	4.5	?
p3	Galaxy S23	4	?

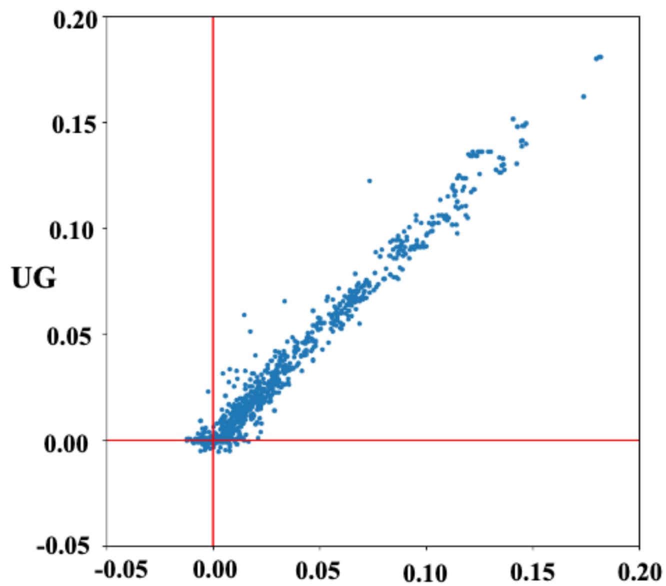
$T_{virtual}$: assuming the target feature can be fully filled.

Feature Importance: Utility Gain gets by $T_{virtual}$

FI(rating) = UG($T_{virtual}$) = Score($T_{virtual}$) - Score(Base)

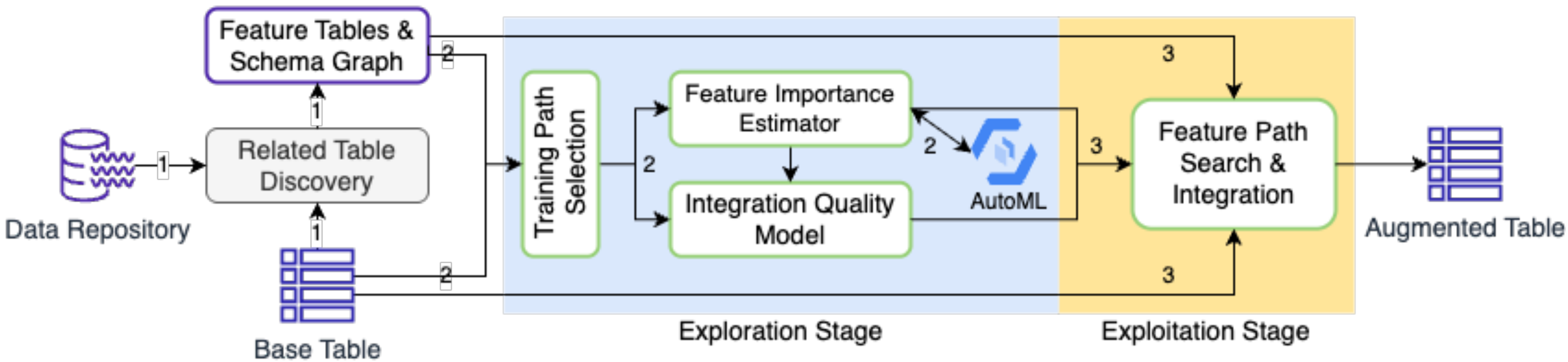
Key Idea: Utility Gain decomposition

$$UG(aug) = IQ(aug) * FI(target\_feature)$$

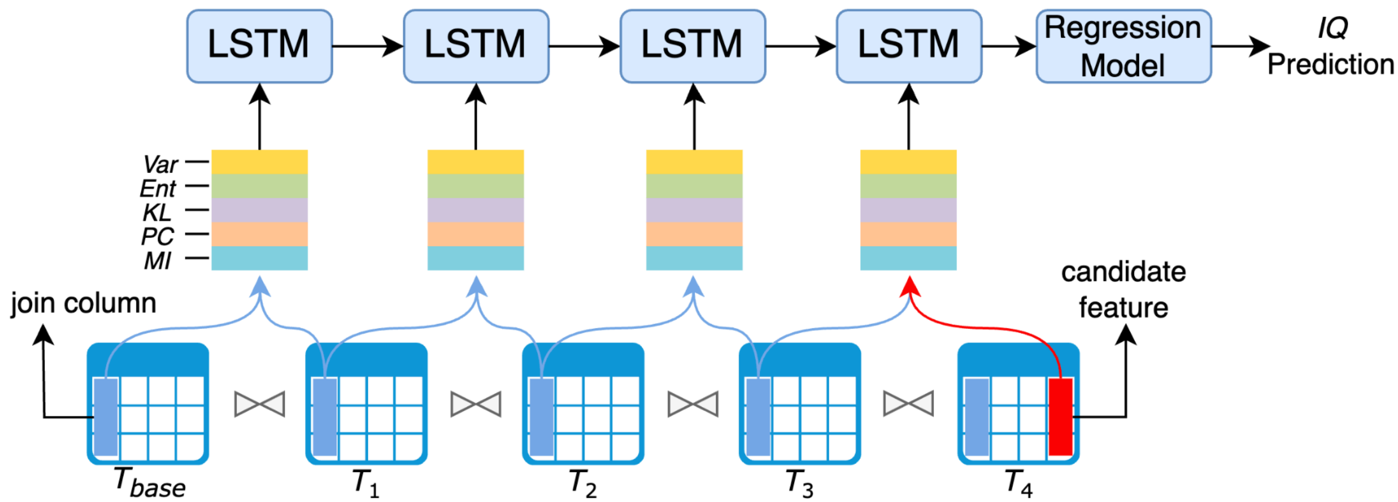


$IQ{\cdot}FI$ and UG

Pipeline Overview



Method: Integration Quality Estimation



Intuition:
Inferencing IQ by pairwise table features and statistics, without join materialization.

- Variance
- Entropy
- KL-divergence
- Pearson-correlation
- Mutual-information

(Inspired by Liu *et al.*, 2022)

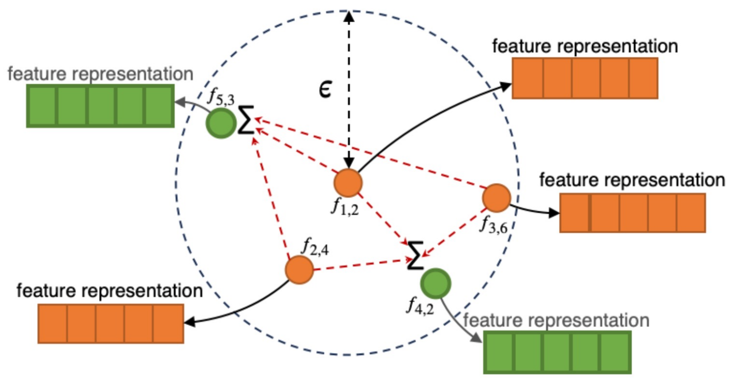
Method: Feature Importance Estimation

Intuition: Similar feature should have similar Feature Importance

1. Feature Clustering
   - column metadata
   - column instances

2. Sample FI data points

3. Estimate unseen features

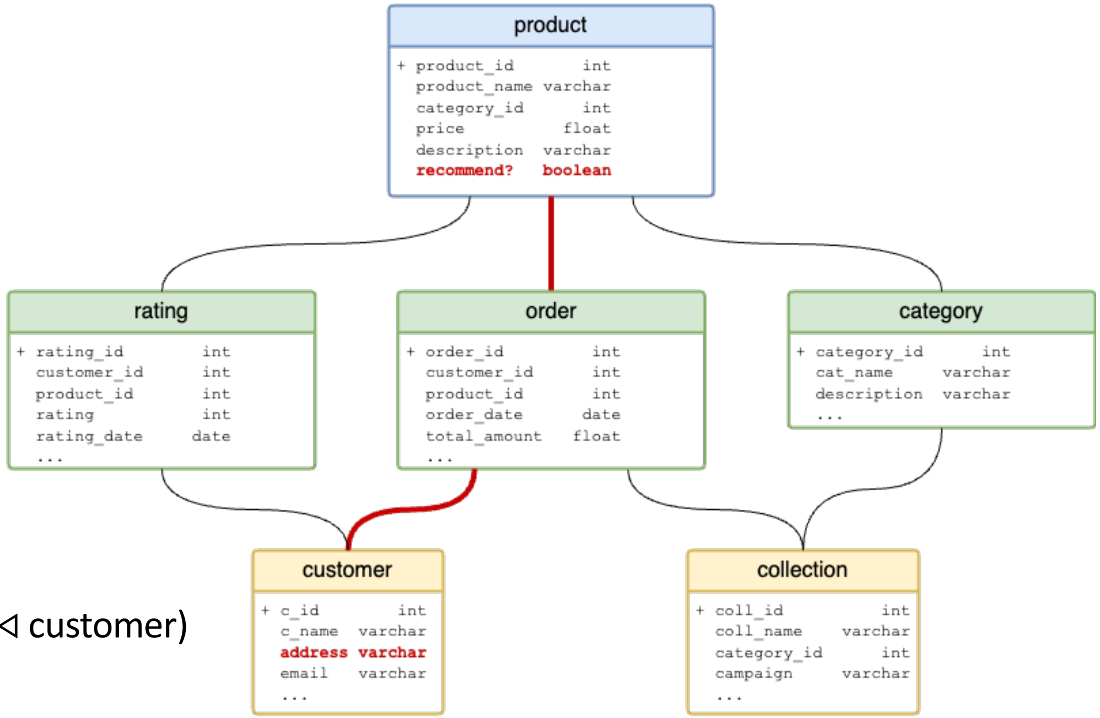


Method: Integration Strategies Search

Pruning Strategies:

- Integration Quality monotonicity

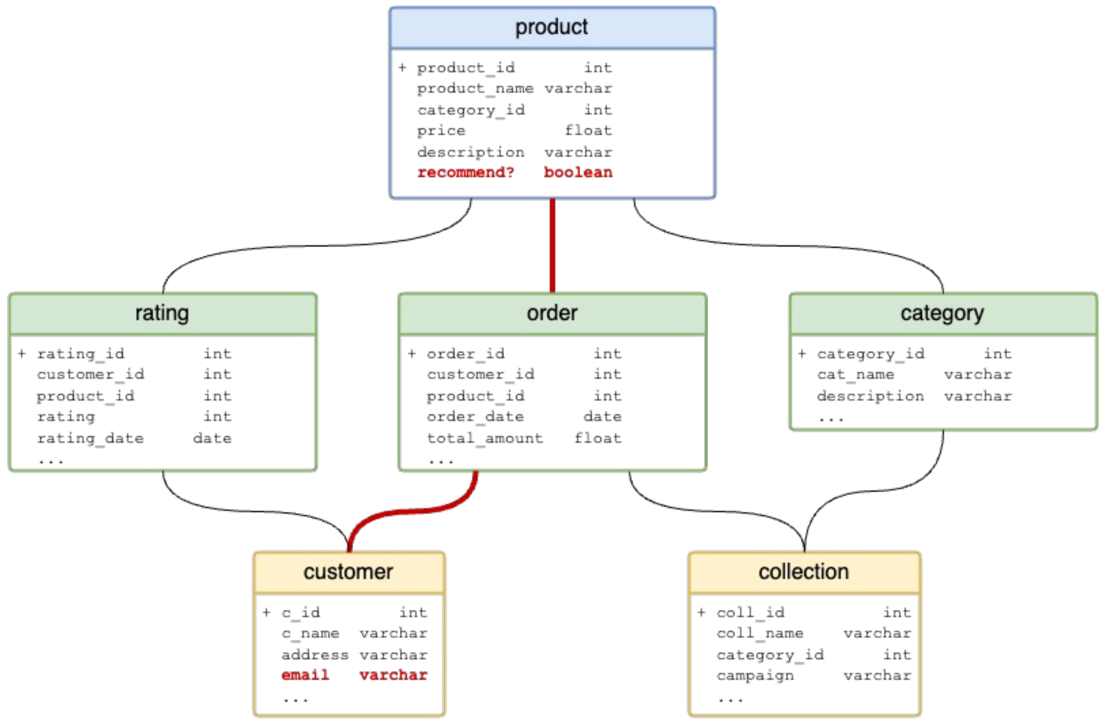
IQ(product ⋈ order) >= IQ(product ⋈ order ⋈ customer)



Method: Integration Strategies Search

Pruning Strategies:

- Integration Quality monotonicity
- Feature Importance lower bound



Experiment Settings

Datasets

	Task	Metrics	# Tables	# Columns	Table Source
School	Classification	Accuracy	121	1,295	NYU Auctus
DonorsChoose	Classification	Accuracy	73	221	Kaggle
Diabetes	Classification	Accuracy	71	204	Kaggle
Fraud Detection	Classification	F1	81	254	Kaggle
Poverty	Regression	MAE	98	408	NYU Auctus
Air	Regression	MSE	75	603	NYU Auctus

Experiment Settings

Baselines

- Exhaustive Search
- J. M. Kanter *et al. (IEEE DSAA*, 2015*)*
- N. Chepurko *et al. (VLDB, 2020)*
- J. Liu *et al. (ICDE, 2022)*
- S. Galhotra *et al. (ICDE, 2023)*
- *A. Ionescu et al. (ICDE, 2024)*

Experiment Results

Datasets	Metrics	Feature Budgets	Methods: Exhaustive	DFS	ARDA	AutoFeature	AutoFeat	METAM	FEATPILOT
School	Accuracy	1	0.704(4)	0.704(4)	0.697(7)	0.708(3)	0.704(4)	0.790(2)	**0.823**(1)
		5	0.730(4)	0.700(7)	0.808(2)	0.704(6)	0.710(5)	0.801(3)	**0.891**(1)
		10	0.704(6)	0.692(7)	0.794(3)	0.723(4)	0.718(5)	0.816(2)	**0.880**(1)
DonorsChoose	Accuracy	1	0.682(4)	0.656(5)	**0.856**(1)	0.708(3)	0.656(5)	0.656(5)	0.822(2)
		5	0.834(4)	0.820(5)	0.890(2)	0.852(3)	0.681(6)	0.659(7)	**0.954**(1)
		10	0.837(5)	0.854(4)	0.901(2)	0.896(3)	0.818(7)	0.820(6)	**0.961**(1)
Diabetes	Accuracy	1	0.521(6)	0.521(6)	0.525(4)	0.585(3)	0.525(4)	0.616(2)	**0.678**(1)
		5	0.740(2)	0.631(5)	0.584(7)	0.649(3)	0.605(6)	0.647(4)	**0.742**(1)
		10	**0.746**(1)	0.647(5)	0.616(7)	0.651(4)	0.618(6)	0.655(3)	0.744(2)
Fraud Detection	F1	1	0.325(4)	0.068(7)	0.416(3)	0.145(6)	0.296(5)	**0.437**(1)	0.435(2)
		5	0.440(3)	0.070(7)	0.422(4)	0.152(6)	0.422(4)	0.446(2)	**0.493**(1)
		10	0.517(2)	0.084(7)	0.450(4)	0.162(6)	0.441(5)	0.464(3)	**0.540**(1)
Poverty	MAE	1	8781.90 (2)	13620.14 (7)	12389.54 (3)	13532.57 (6)	12944.16(4)	13077.66 (5)	**8222.34** (1)
		5	7373.94 (2)	13410.07 (6)	12389.54 (3)	13532.57 (7)	12558.93(4)	12956.29 (5)	**7322.44** (1)
		10	7309.52 (2)	13077.66 (6)	12164.23 (4)	13411.85 (7)	11213.23(3)	12786.82 (5)	**7182.38** (1)
Air	MSE	1	1.201 (7)	1.184 (5)	**0.969** (1)	1.259 (6)	1.061(4)	1.101 (2)	1.101 (2)
		5	0.983 (5)	0.985 (6)	0.793 (2)	1.219 (7)	0.915(4)	0.900 (3)	**0.762** (1)
		10	0.873(5)	0.943 (6)	0.761 (2)	1.202 (7)	0.820(4)	0.762 (3)	**0.715** (1)