

AutoFeat: Transitive Feature Discovery over Join Paths



**Andra
Ionescu**



**Kiril
Vasilev**



**Florena
Buse**



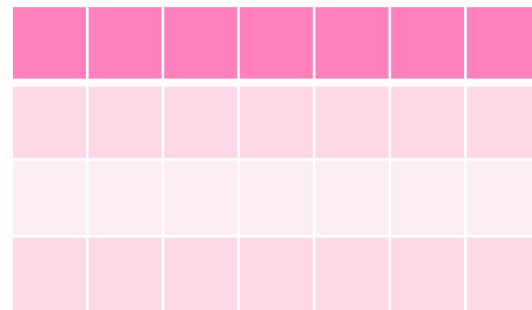
**Rihan
Hai**

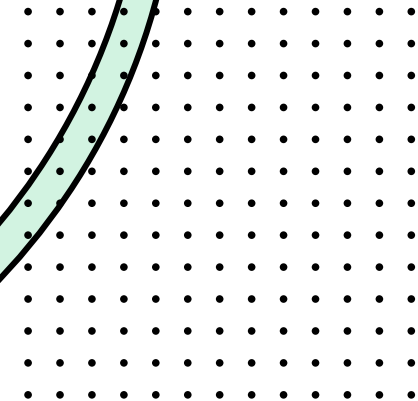
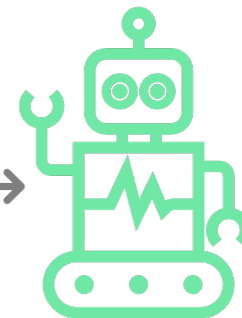


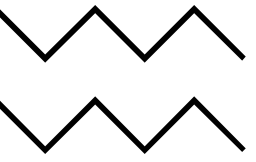
**Asterios
Katsifodimos**

PROBLEM

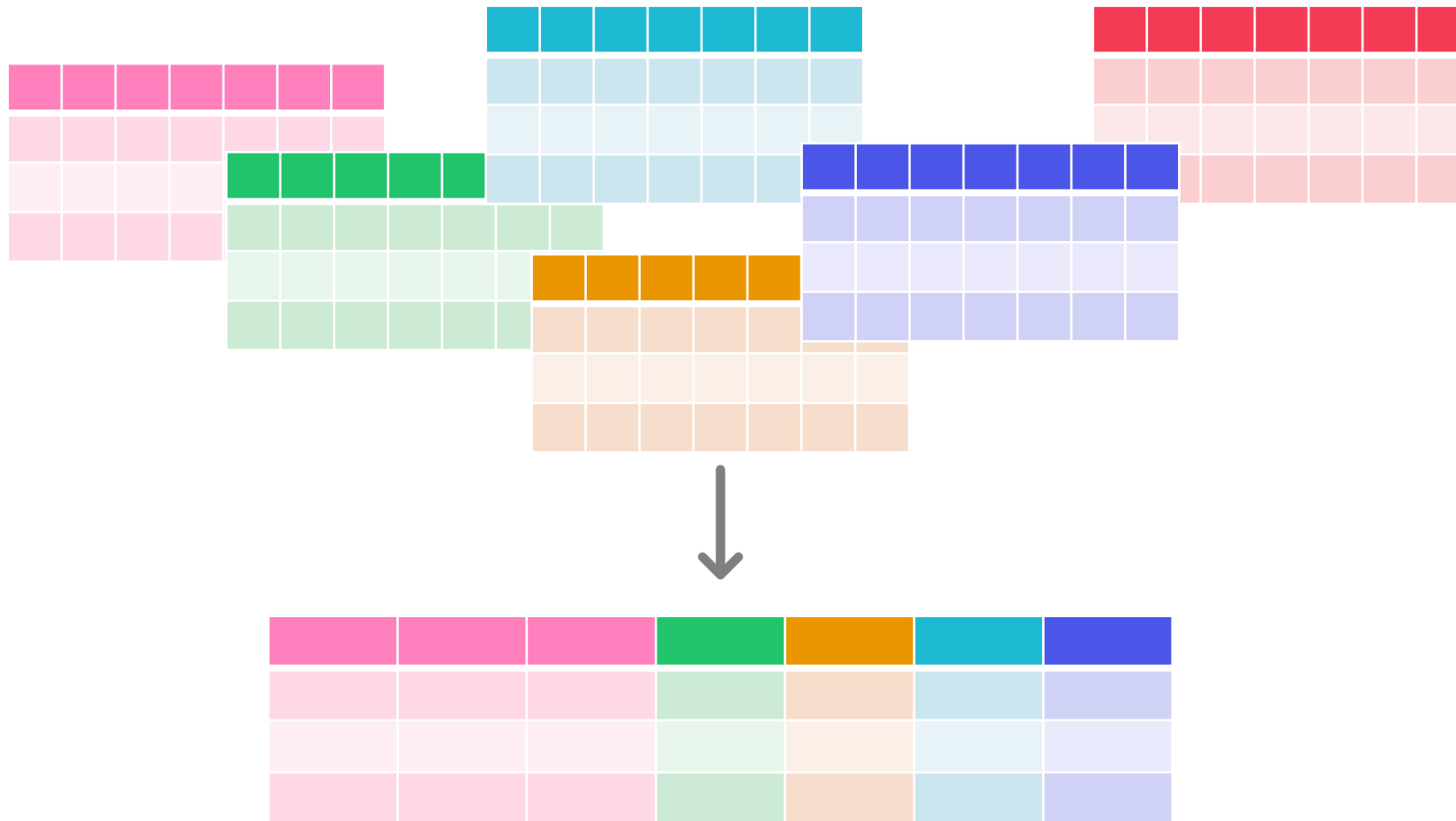
Input data of an ML model is a single table

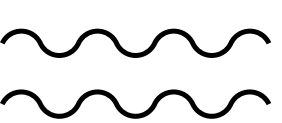






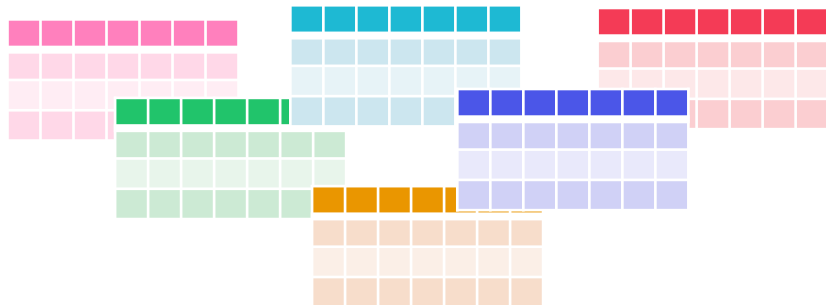
Input dataset is the result of data augmentation and feature selection



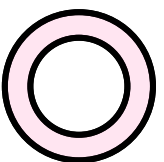
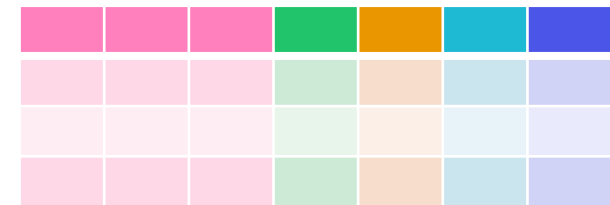


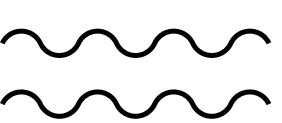
Dataset Augmentation

Collection of datasets



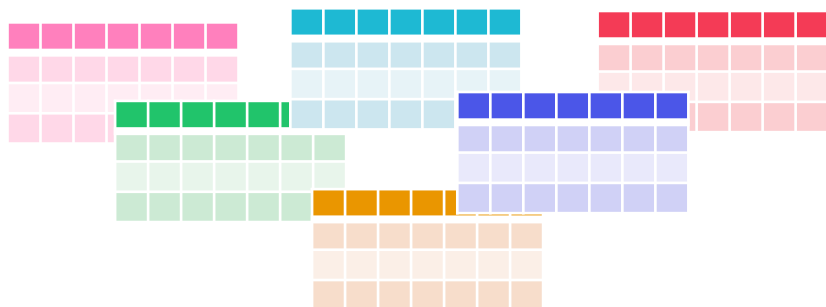
Training dataset



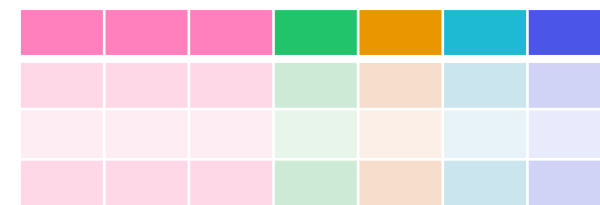


Dataset Augmentation

Collection of datasets

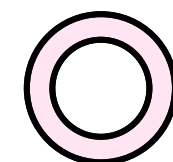


Training dataset



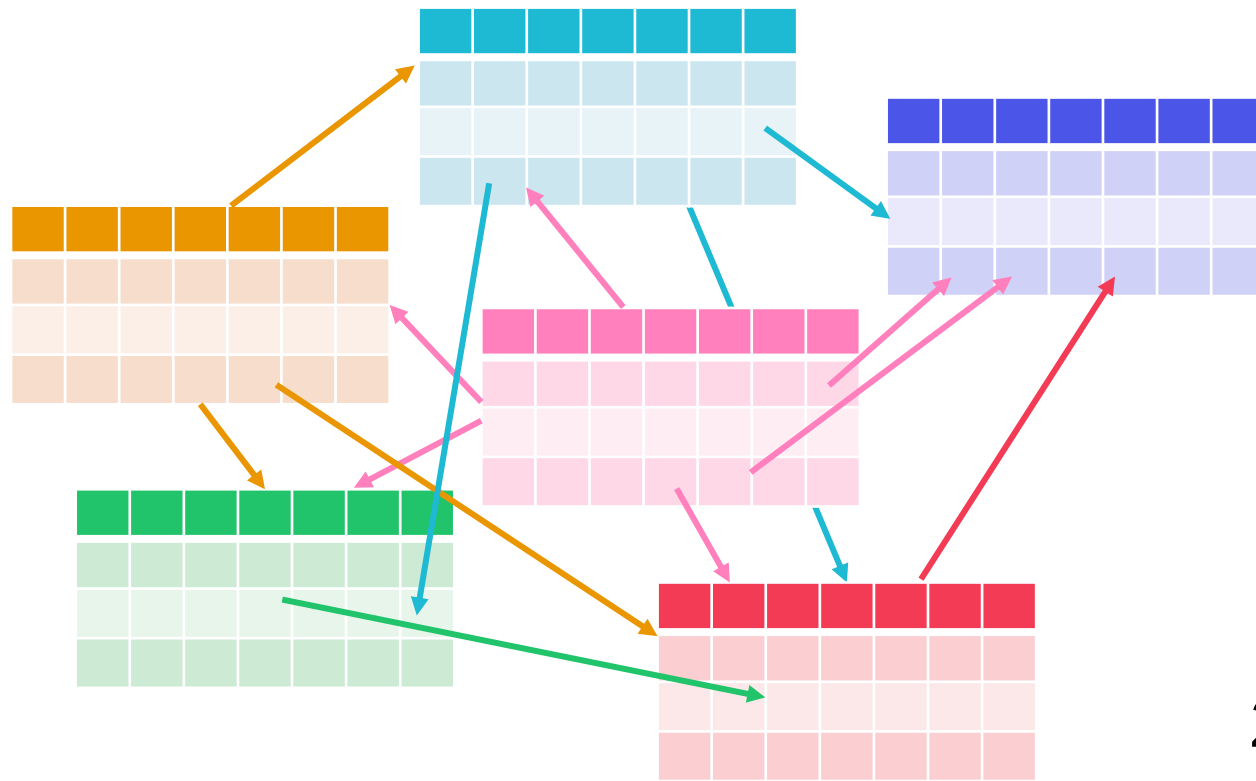
When PK-FK are known:

1. Search for datasets
2. Join datasets
3. Apply feature selection



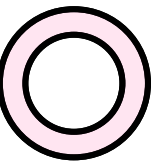


Dataset Augmentation



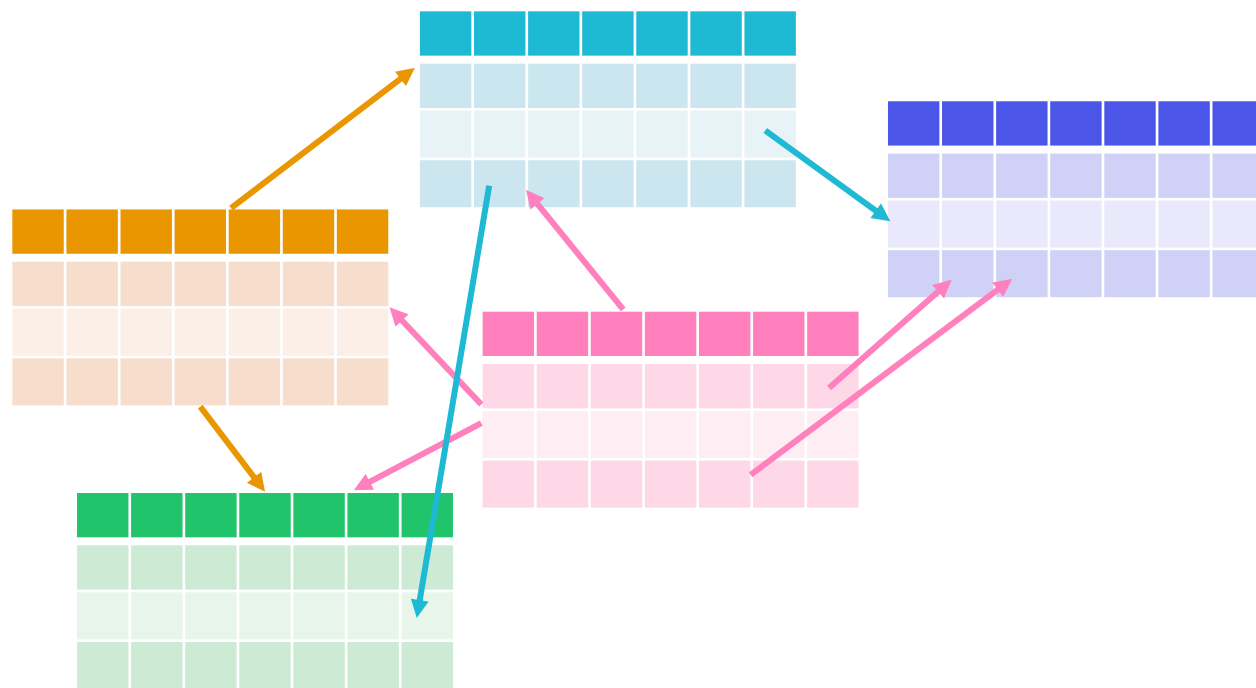
When PK-FK are missing:

1. Dataset discovery
2. Join data
3. Apply feature selection

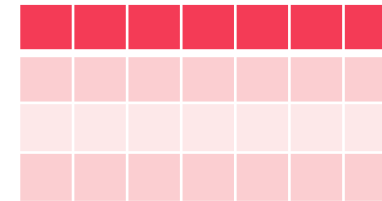




Dataset Augmentation

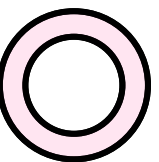


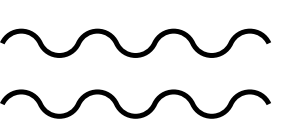
- Spurious relations



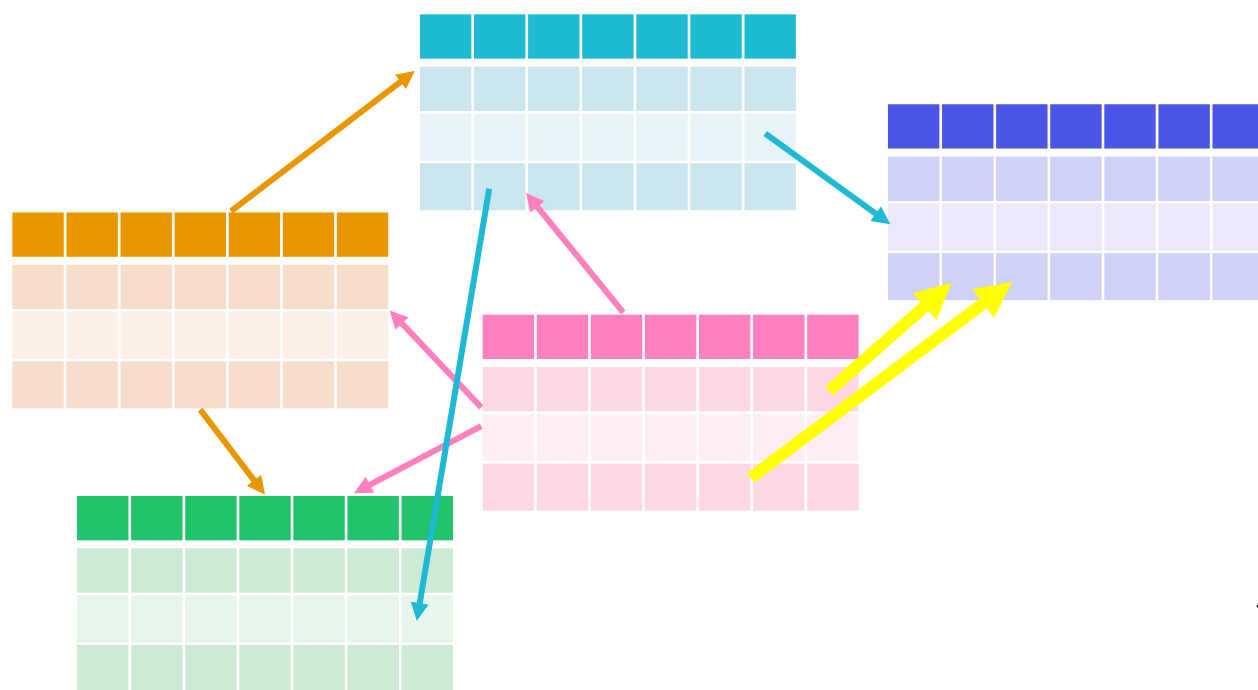
When PK-FK are missing:

1. Dataset discovery
2. Join data
3. Apply feature selection





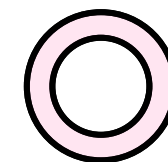
Dataset Augmentation



- Multiple join columns

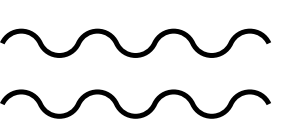
When PK-FK are missing:

1. Dataset discovery
2. Join data
3. Apply feature selection

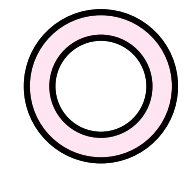
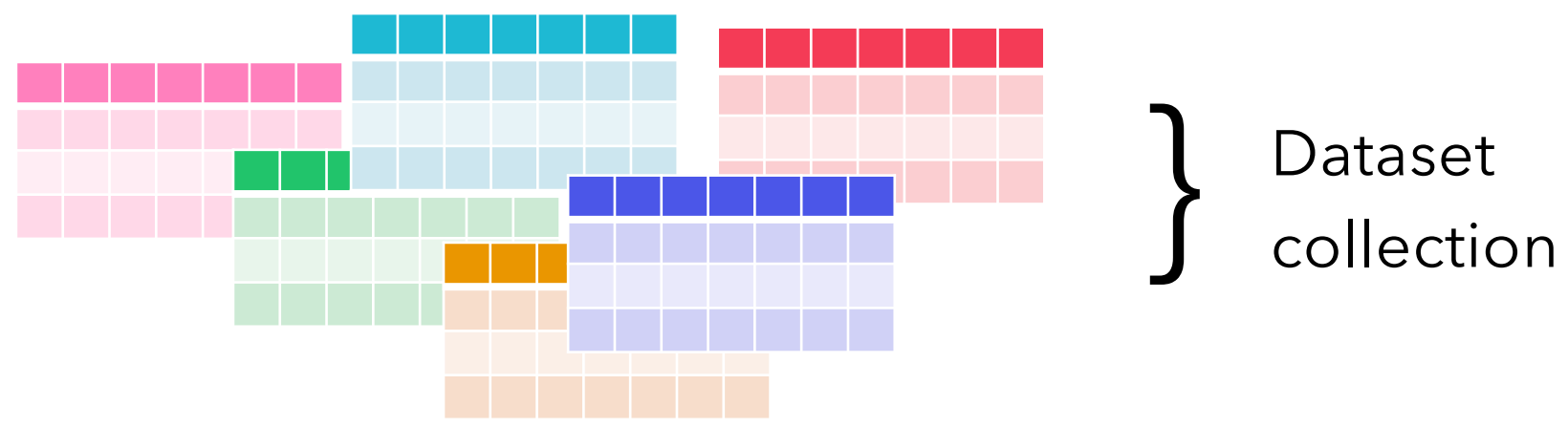
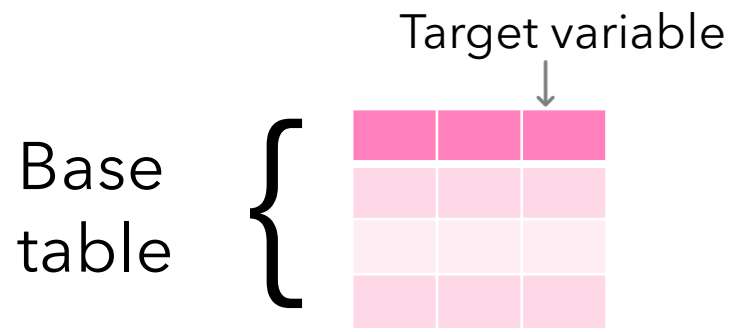


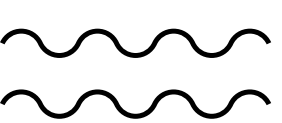


FEATURE DISCOVERY

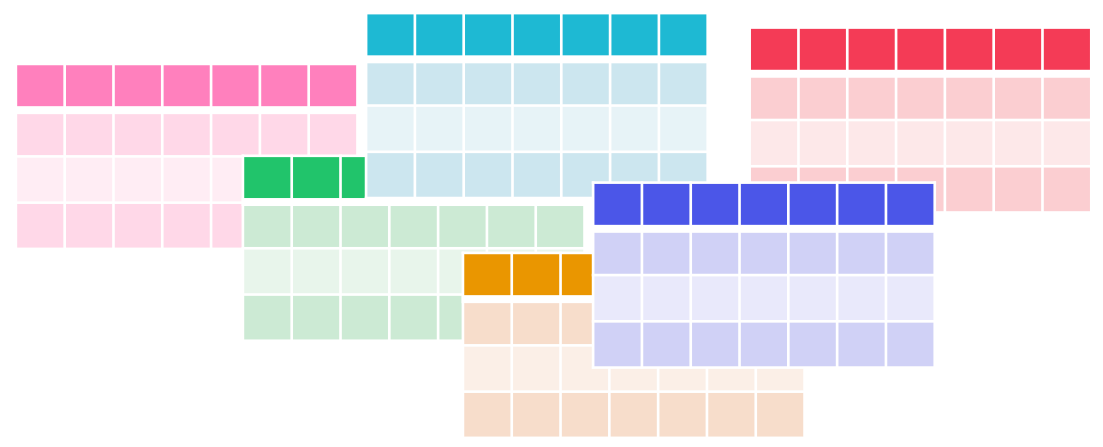
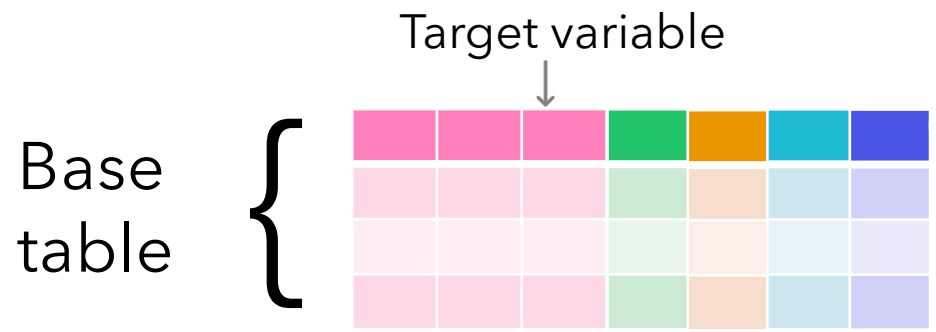


Feature Discovery

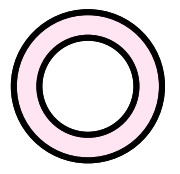


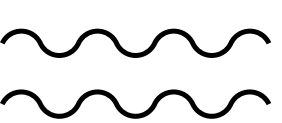


Feature Discovery

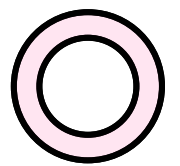
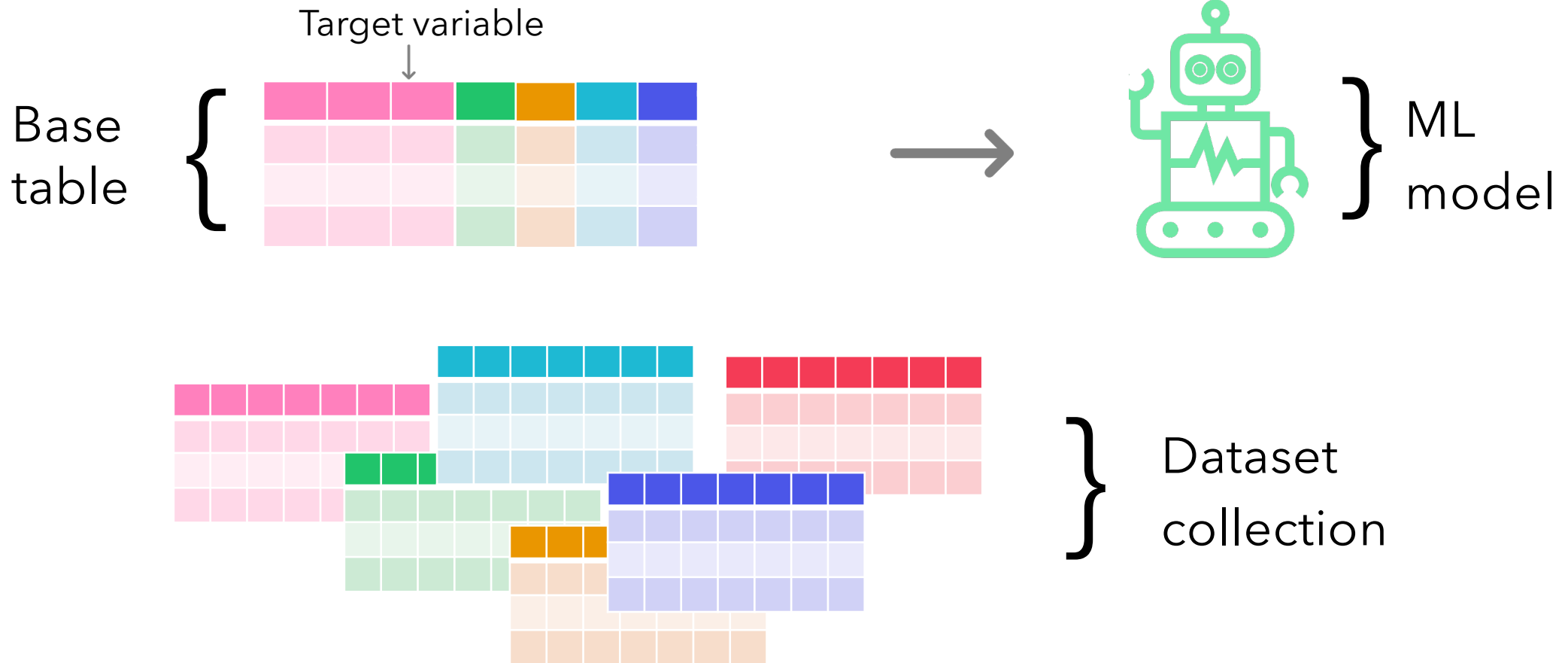


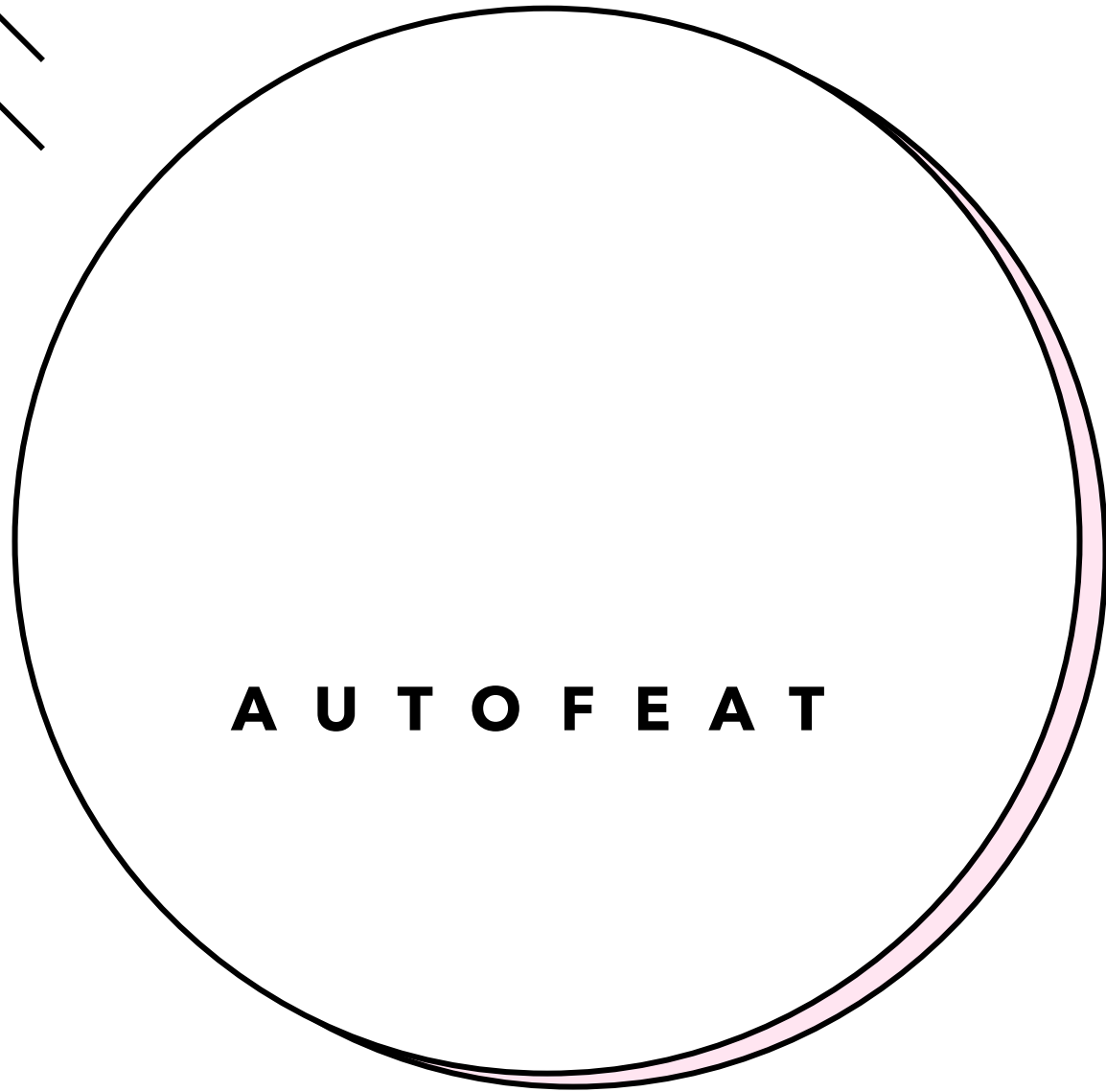
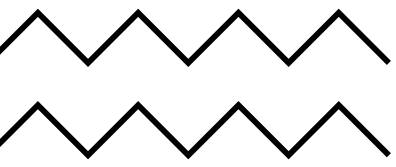
Dataset collection



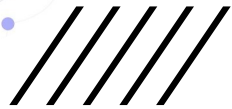


Feature Discovery





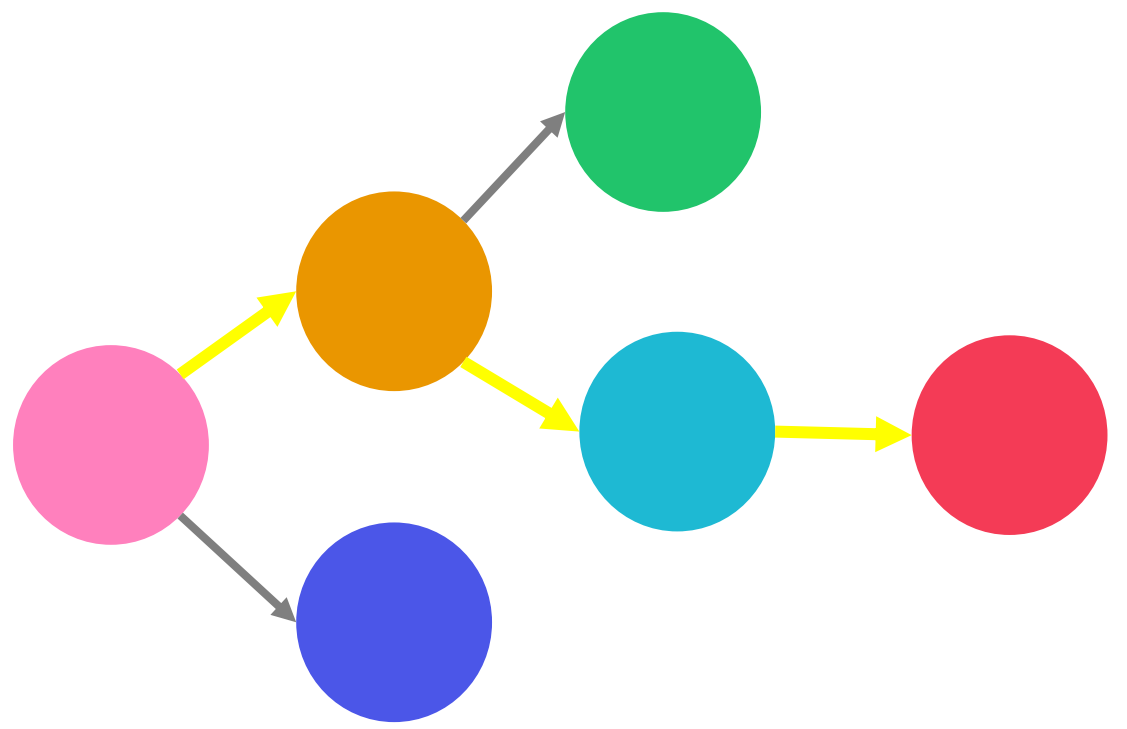
A U T O F E A T





AutoFeat

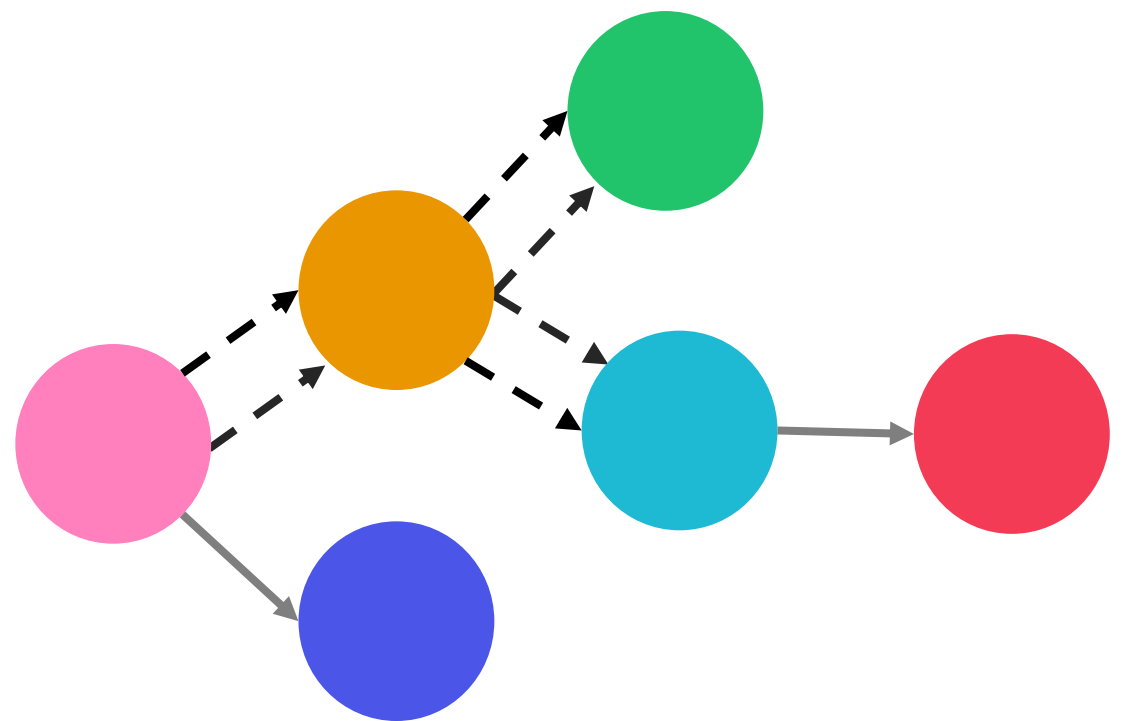
- Join-path length:
 - ✓ single-hop
 - ✓ multi-hop





AutoFeat

- Joinability graph
 - ✓ simple graph
 - ✓ multi-graph





AutoFeat

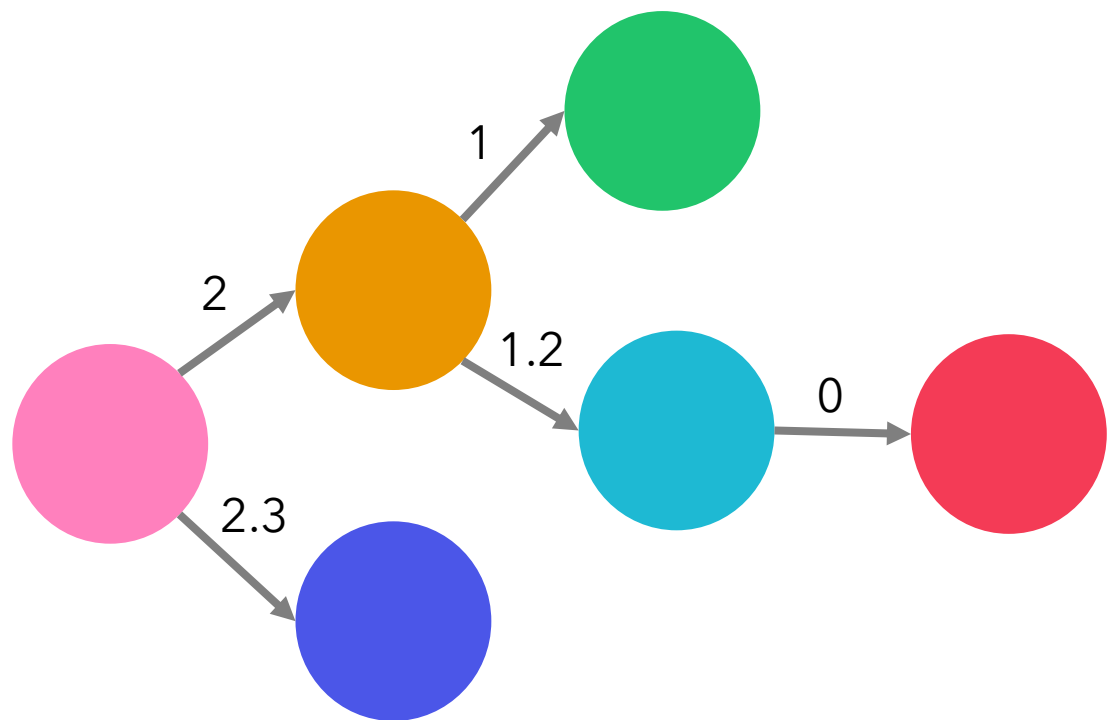
• Path / Feature selection:

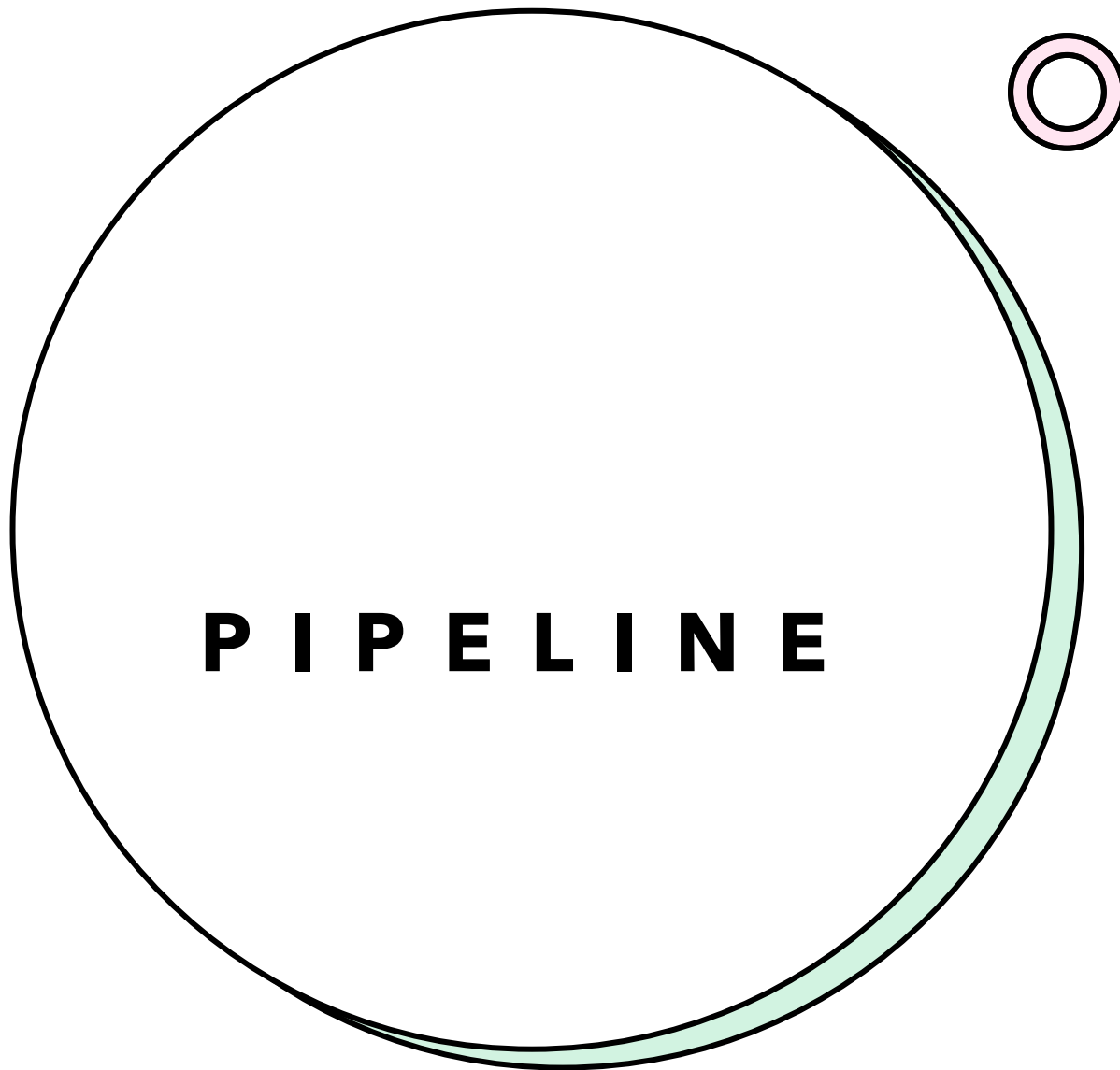


ranking-based



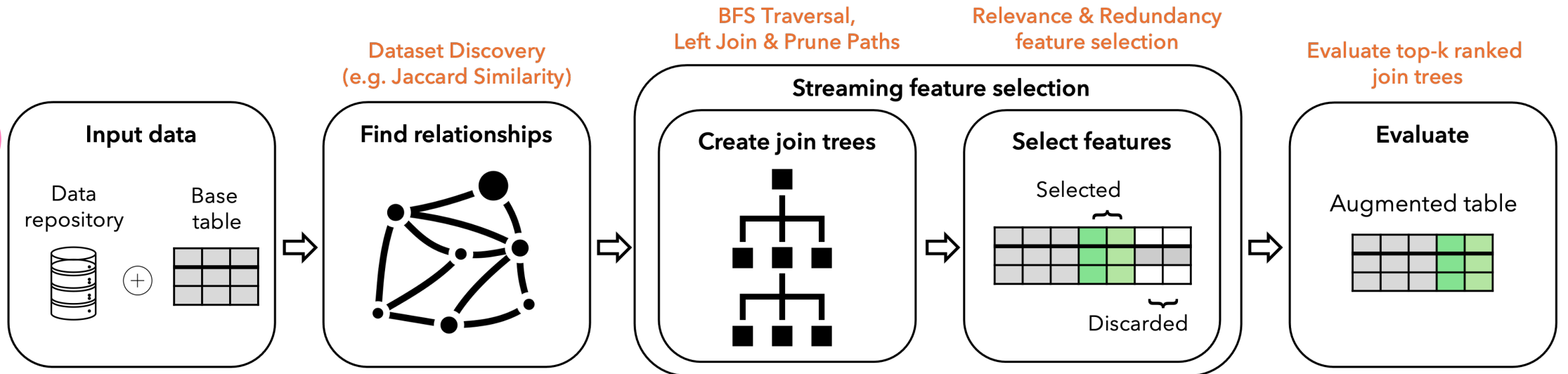
model-execution based



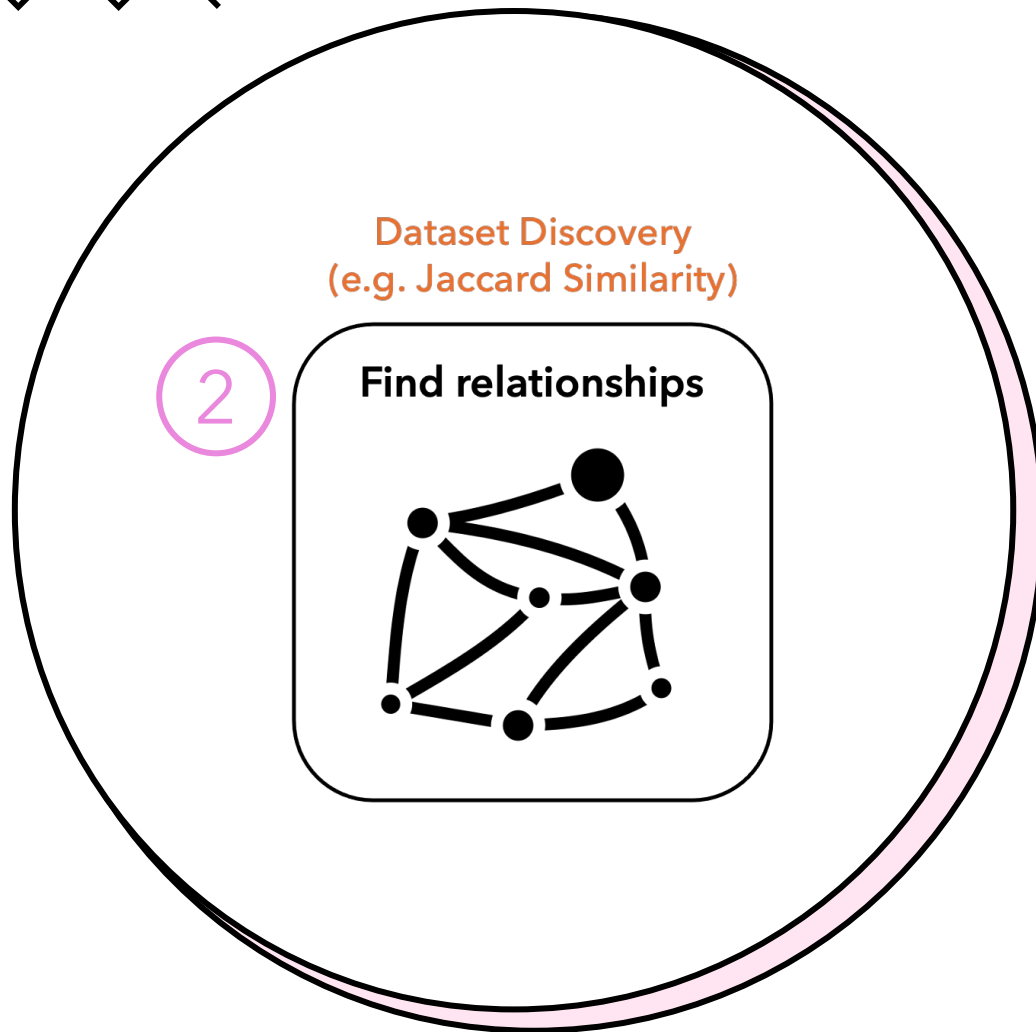


AutoFeat Pipeline

1



Dataset Relation Graph



Dataset Discovery

- Valentine - schema matching tool suite [1]

DRG - weighted graph

- Nodes → Tables
- Edges → Relationships
 - Weight = 1 (PK-FK)
 - Weight = similarity score

[1] Christos Koutras, et al. "Valentine: Evaluating matching techniques for dataset discovery." 2021 ICDE

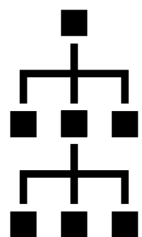
3

BFS Traversal,
Left Join & Prune Paths

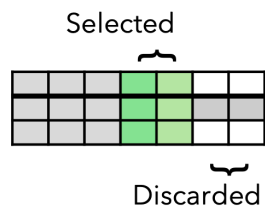
Relevance & Redundancy
feature selection

Streaming feature selection

Create join trees



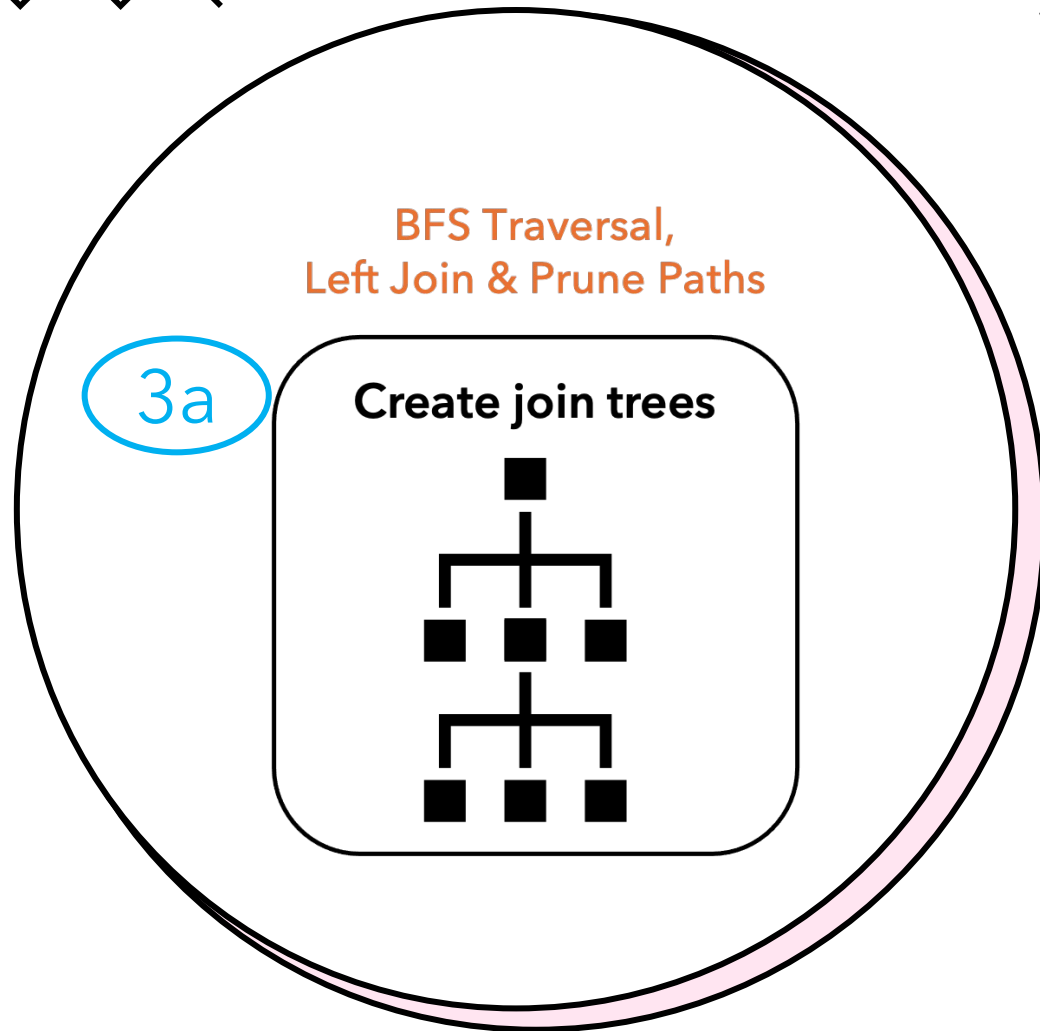
Select features



STREAMING FEATURE SELECTION

FEATURES ARRIVE IN A
STREAMING FASHION WITH
EVERY JOIN

Join Trees



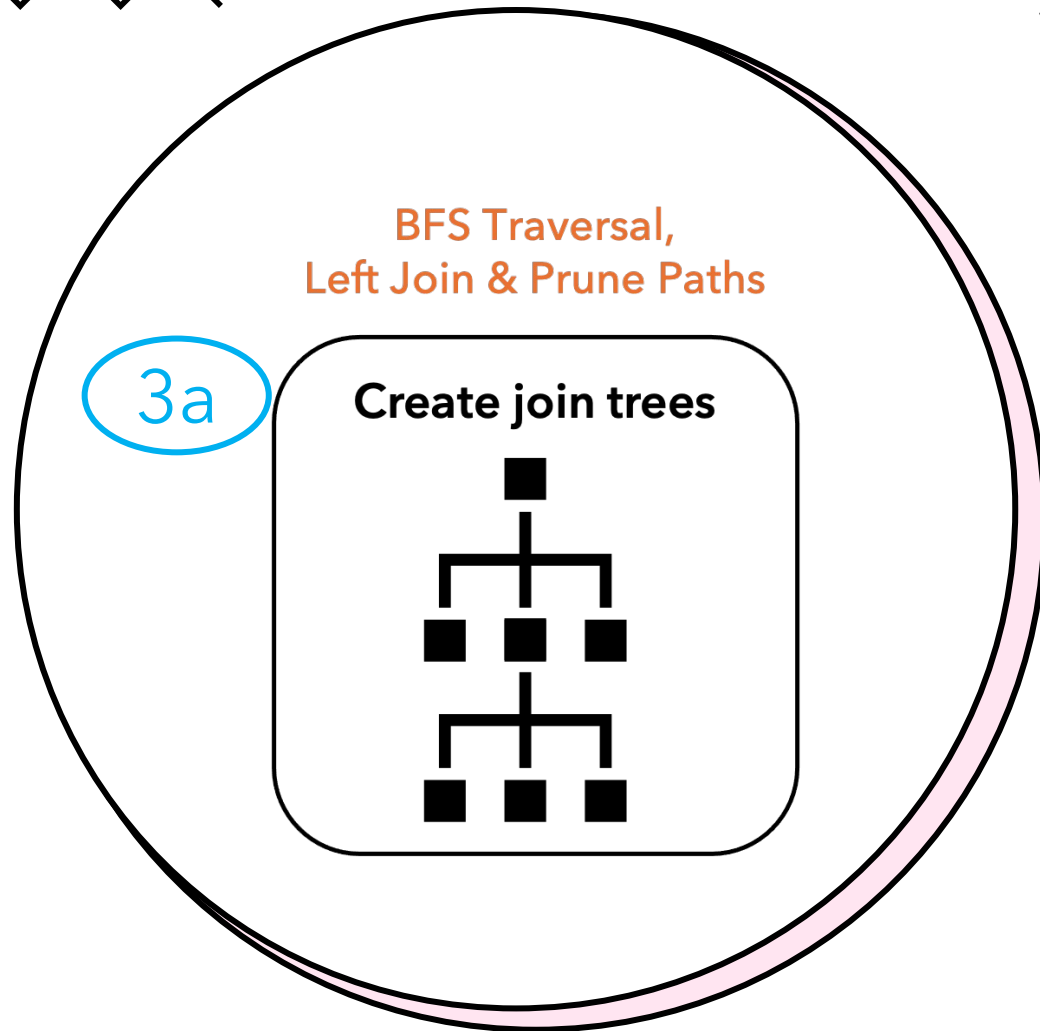
Graph traversal

- Breadth First Search (BFS)
- Evaluate data quality after each level
- Easier error management

Join type

- Left join
- Preserve number of rows
- Avoid introducing class imbalance

Join Trees



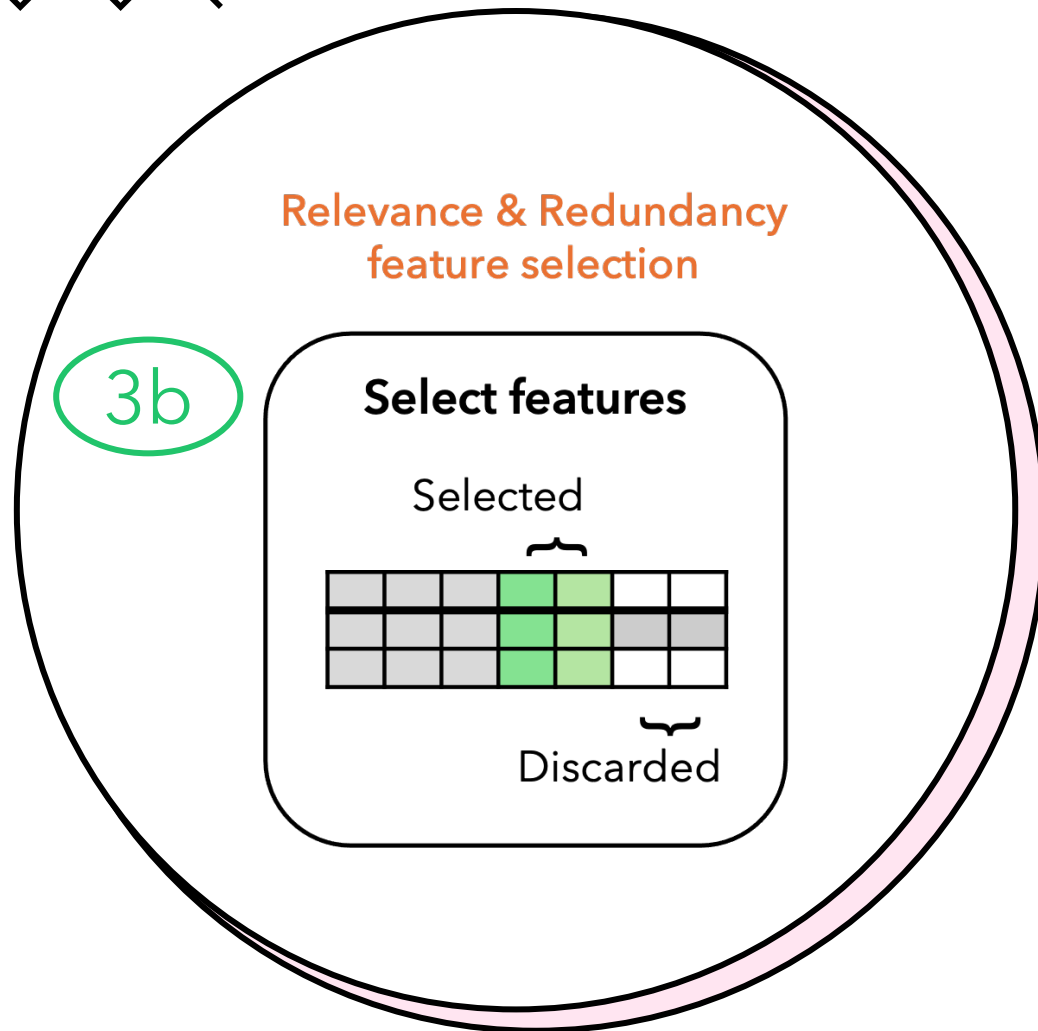
Join paths

- Sequence of edges
- Chain of joins

Prune paths

- Similarity score
- Data quality - null values ratio

Feature Selection



Relevance

- Spearman correlation - rank correlation

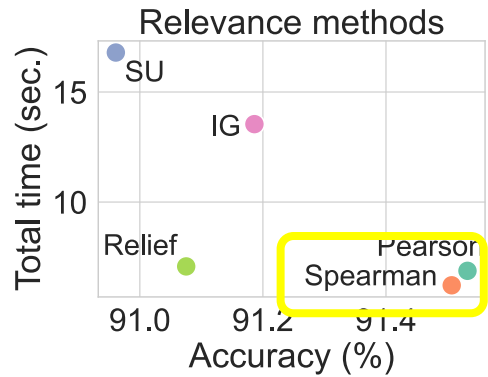
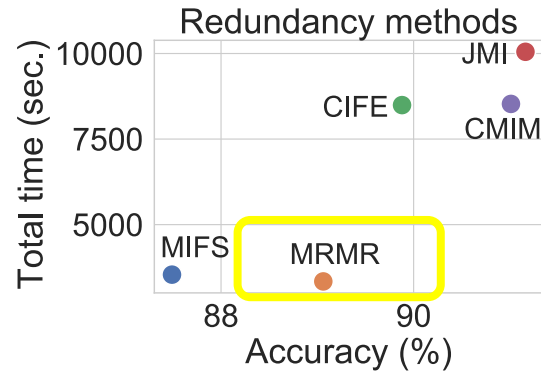
Redundancy

- MRMR - with more selected features, the effect of redundancy is reduced

Ranking

- Linear function of relevance and redundancy scores

Feature Selection



Relevance

Information Gain

Pearson correlation

Spearman correlation

Relief

Redundancy

Mutual Information Feature Selection

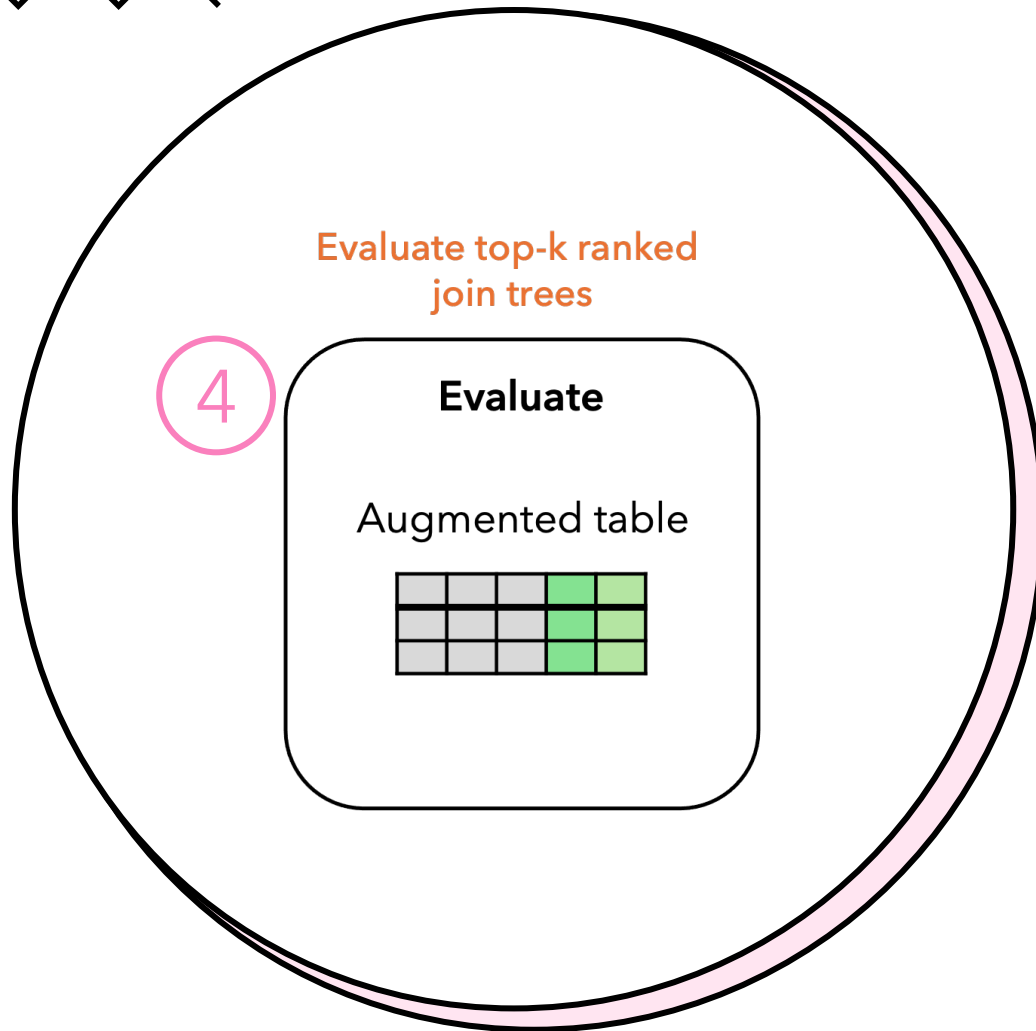
Minimum Redundancy Maximum Relevance

Conditional Infomax Feature Extraction

Join Mutual Information

Conditional Mutual Information Maximisation

Evaluate Join Trees

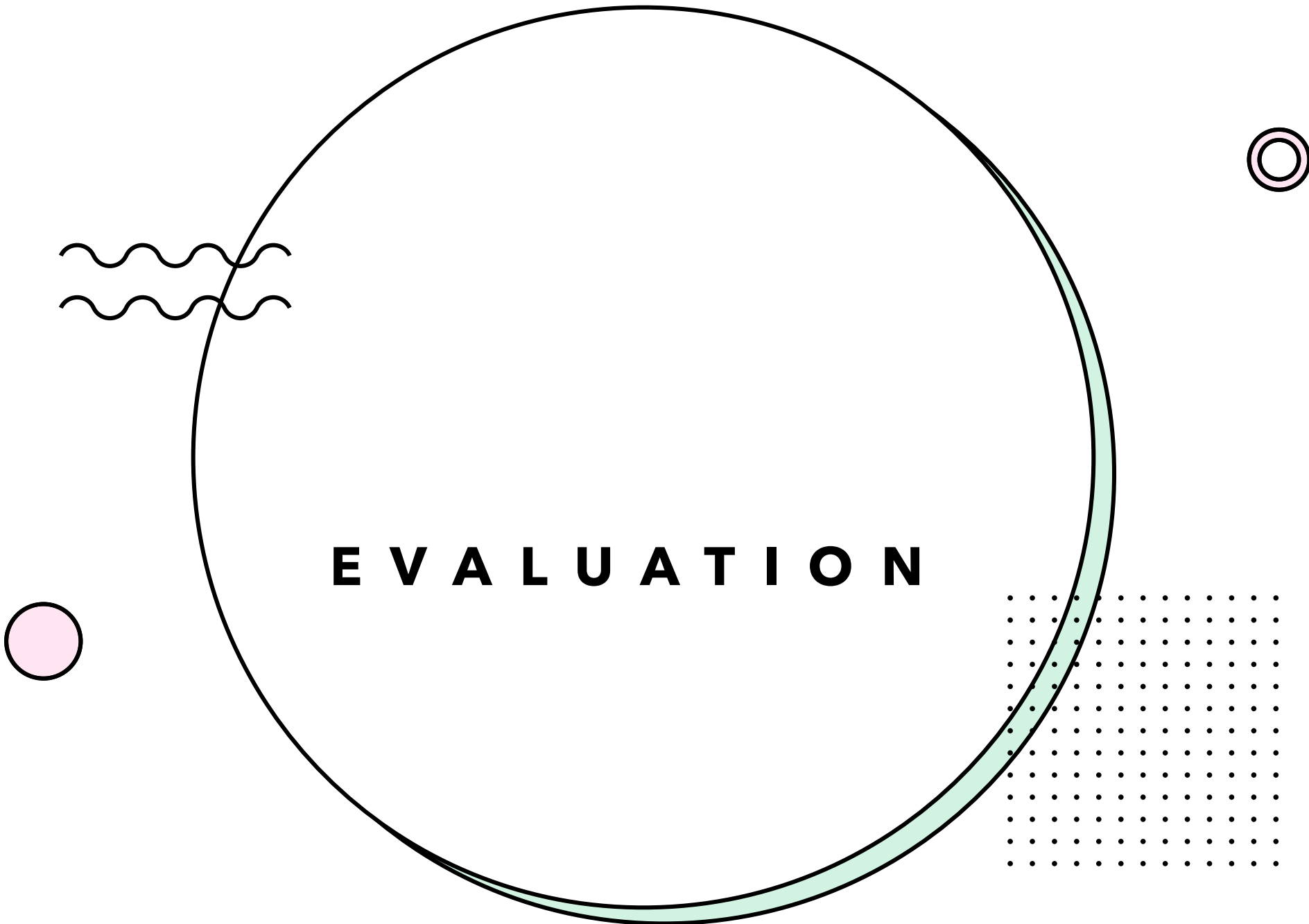


Top-k join trees

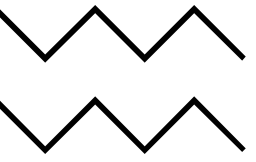
- Based on the ranking

Augment Base Table

- Train ML model



EVALUATION



Setup

Datasets

7 OpenML

1 SOTA

ML models

Decision trees
from AutoGluon

Metrics

Efficiency

Effectiveness





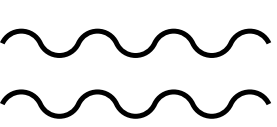
Baselines

Base	<ul style="list-style-type: none">• Non-augmented base table
Join All	<ul style="list-style-type: none">• Join all tables
Join All + FS	<ul style="list-style-type: none">• Join all, then apply feature selection
ARDA [2]	<ul style="list-style-type: none">• Random Injection of noise
Multi-Armed Bandit [3]	<ul style="list-style-type: none">• Exploration - Exploitation strategy

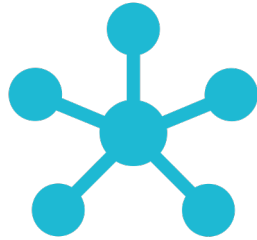
[2] Nadiia Chepurko, et al. "ARDA: Automatic Relational Data Augmentation for Machine Learning." 2020 VLDB

[3] Jiabin Liu, et al. "Feature augmentation with reinforcement learning." 2022 ICDE





Scenarios



Known Relations

Known PK-FK connections

Star/Snowflake schema

Reproduce the results from baselines

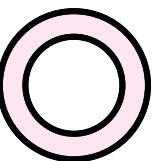


Discovered Relations

Unknown PK-FK connections

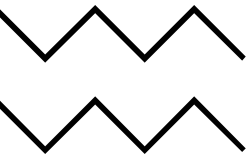
Dense multi-graph

Show the predictive power of AutoFeat



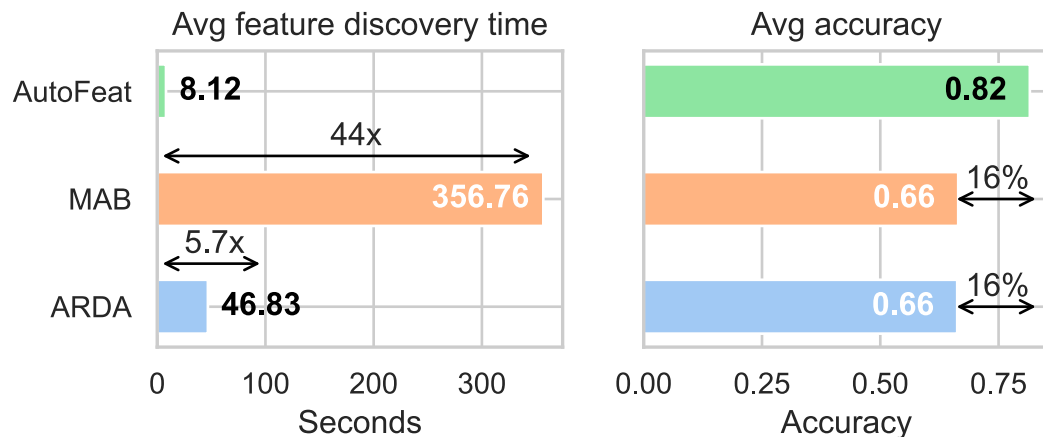


R E S U L T S



KNOWN RELATIONS

16% AVERAGE INCREASE IN ACCURACY ACROSS ALL DATASETS AND MODELS



STAR SCHEMATA

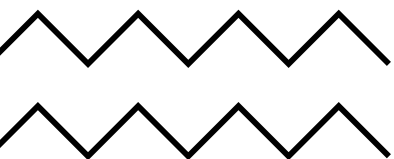
AutoFeat prunes out all the irrelevant tables

SNOWFLAKE SCHEMATA

AutoFeat explores the graph of connections in depth



DISCOVERED RELATIONS



Path analysis

AutoFeat explores the join space in depth
Prunes out irrelevant tables



Effectiveness

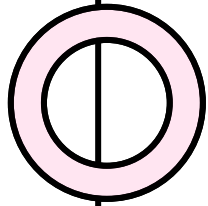
AutoFeat shows increased accuracy from the base table



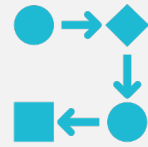
Efficiency

10x faster than MAB
3x faster than ARDA





Conclusion



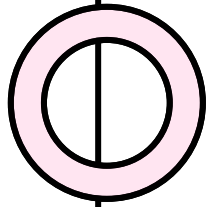
AutoFeat is a more efficient and effective method for automatic feature discovery over long join paths.



AutoFeat works with both star and snowflake schema.



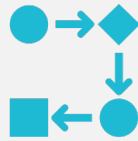
AutoFeat decouples the model training step from feature discovery process
AutoFeat relies on heuristics to prune out irrelevant tables and features.



Thank you!

Open for
work

a.ionescu-3@tudelft.nl



AutoFeat is a more efficient and effective method for automatic feature discovery over long join paths.



AutoFeat works with both star and snowflake schema.



AutoFeat decouples the model training step from feature discovery process
AutoFeat relies on heuristics to prune out irrelevant tables and features.

<https://github.com/delftdata/autofeat>