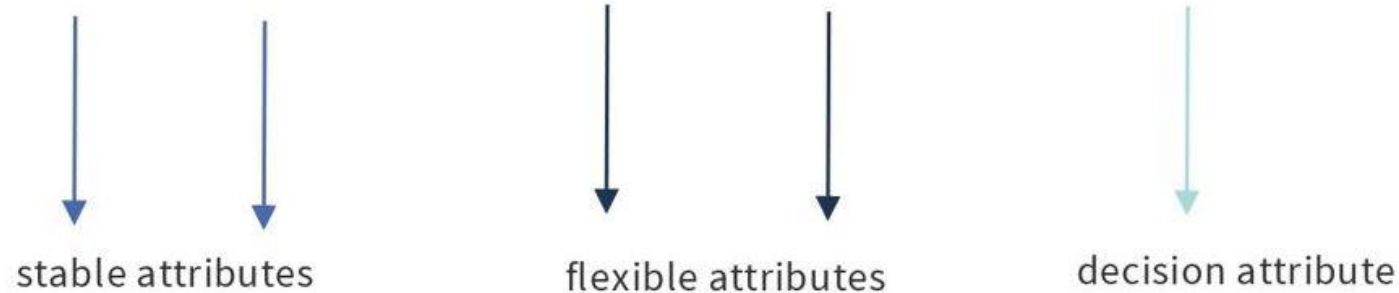


ACTION RULES DISCOVERY:

Leveraging Attributes Correlation Based Vertical Partitioning

ANATOMY OF AN ACTION RULE

$$r = [[(A, a_1) \wedge (G, g_2) \wedge (B, b_1 \rightarrow b_2) \wedge (H, \rightarrow h_2)] \Rightarrow (D, d_1 \rightarrow d_2)]$$

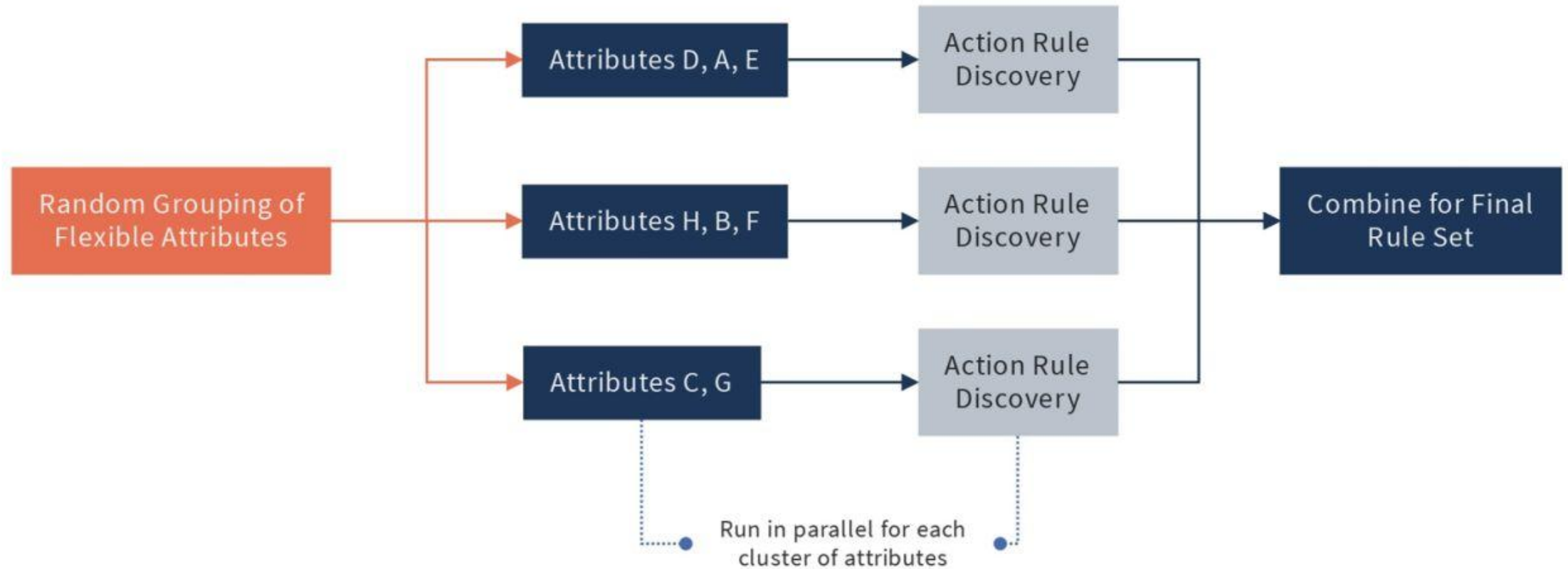


A way to **reclassify objects** from one target value to another by determining which values of flexible attributes can be changed. They are **actionable** and can be used to create suggestions for reclassifying objects.

However, the generation of action rules can be both **time-consuming and computationally intensive**, especially for large datasets.

DISTRIBUTED ACTION RULE DISCOVERY I

Based on **Random** Vertical Data Partitioning

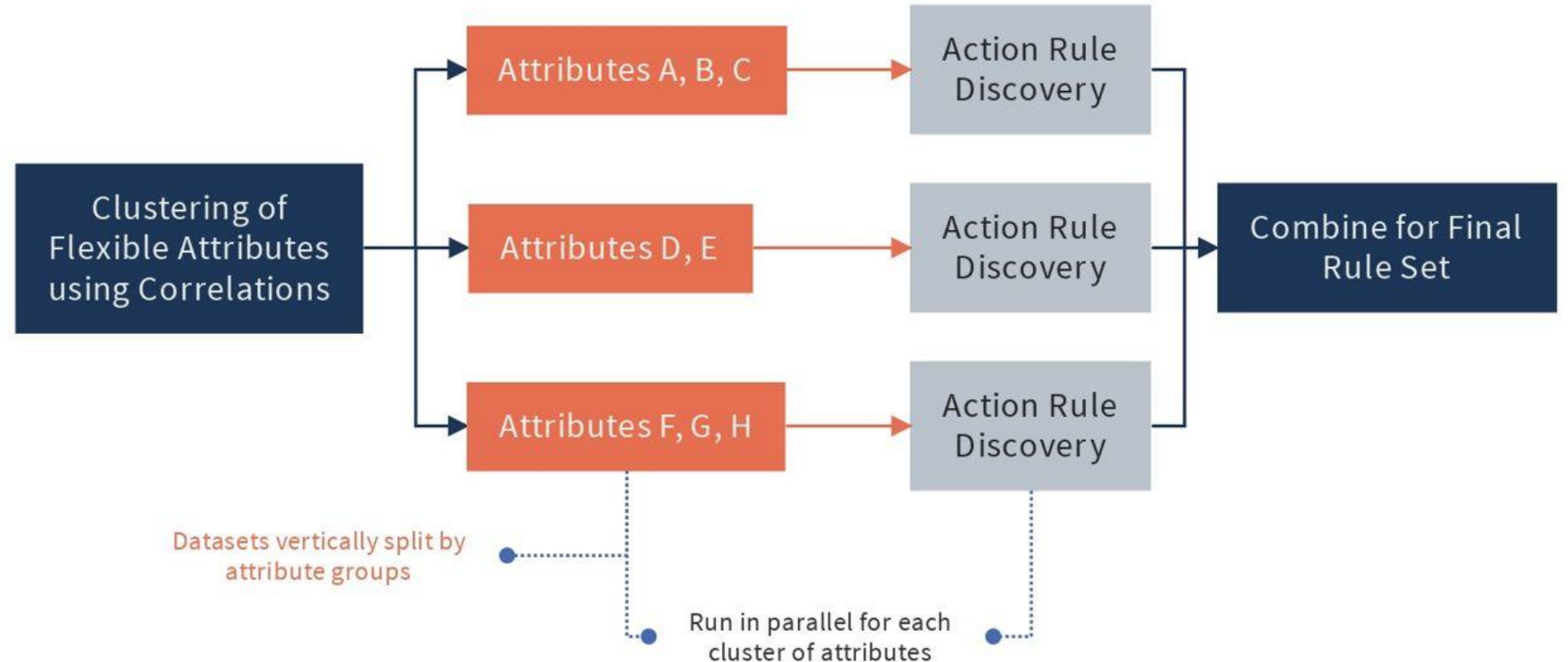


DISTRIBUTED ACTION RULE DISCOVERY II

Based on **Attributes Correlation** and Vertical Data Partitioning

GOAL

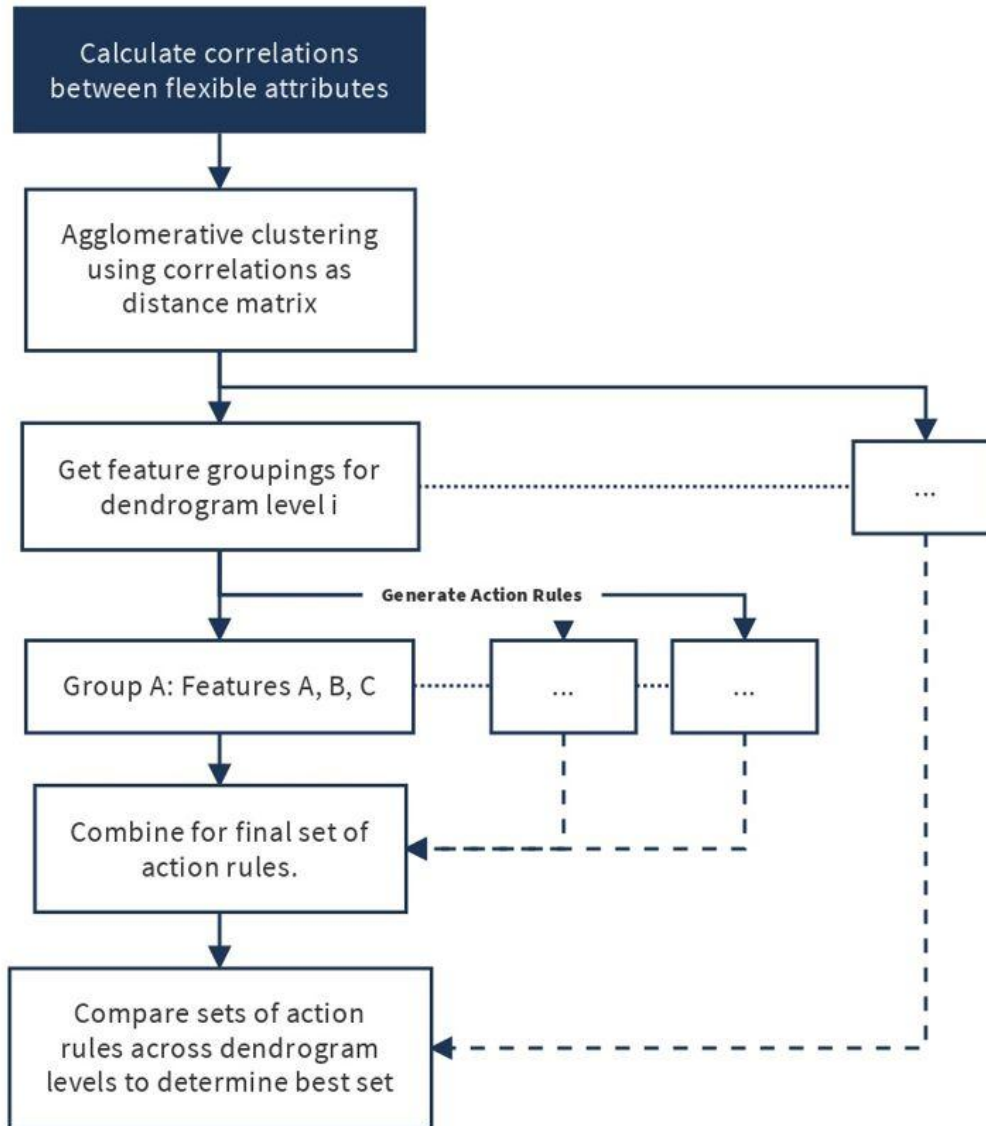
Improve run time for datasets with large numbers of flexible attributes.



Digging deeper into

THE METHODOLOGY



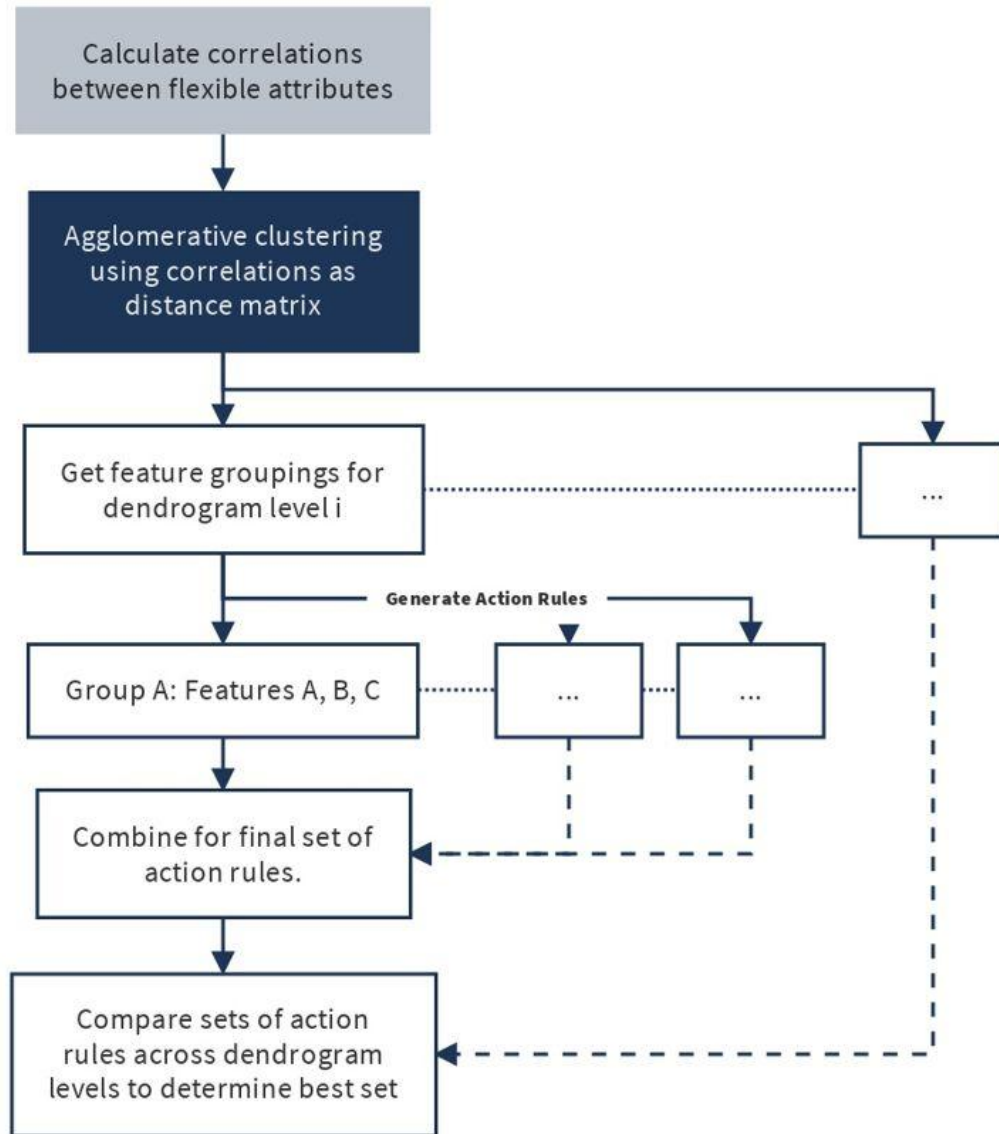


CALCULATE CORRELATIONS BETWEEN FLEXIBLE ATTRIBUTES.

Pearson's correlation coefficient (r) is used for pairs of numerical attributes.

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}}$$

<http://shakedzy.xyz/dython/>



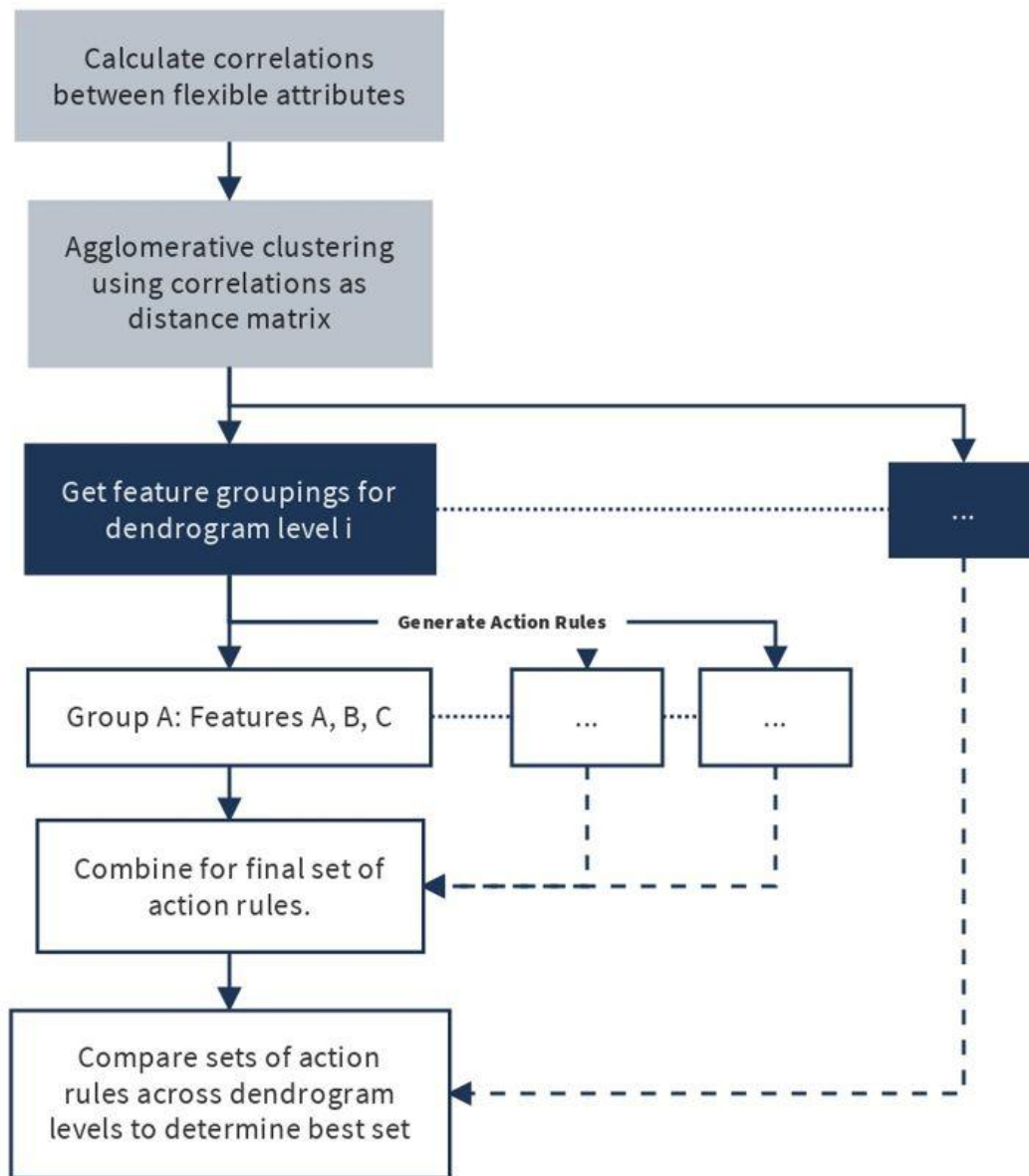
AGGLOMERATIVE CLUSTERING USING CORRELATIONS AS DISTANCE MATRIX

$$DistanceMatrix = 1 - \frac{|associations| + 1}{2}$$

Where **associations** are the feature correlations obtained in the previous step.

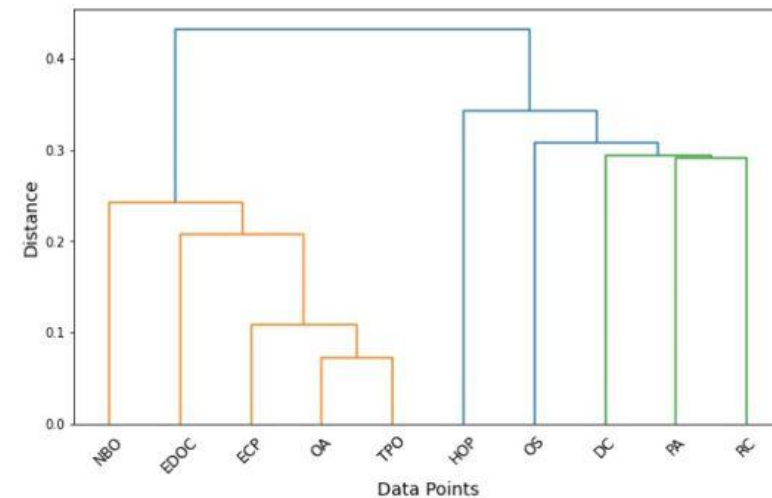
The SciPy library is used to perform the agglomerative clustering.

<https://scipy.org/>



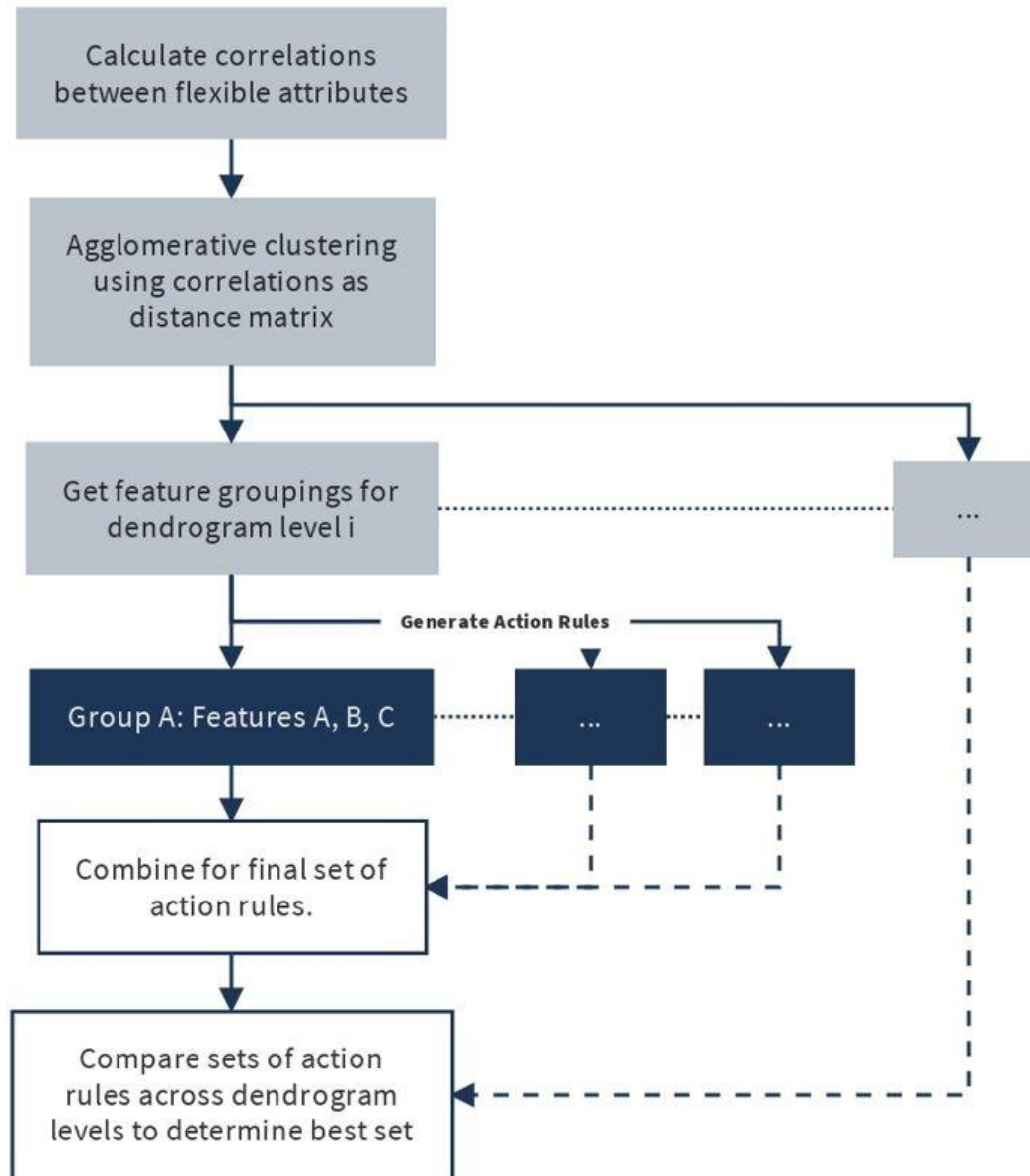
GET FEATURE GROUPINGS FOR DENDROGRAM LEVEL I

Where i is each level of the dendrogram, which we iterate through. Each level contains clusters of features (example below).



Each of these levels can be run in parallel. Ray is used to help build distributed systems in Python.

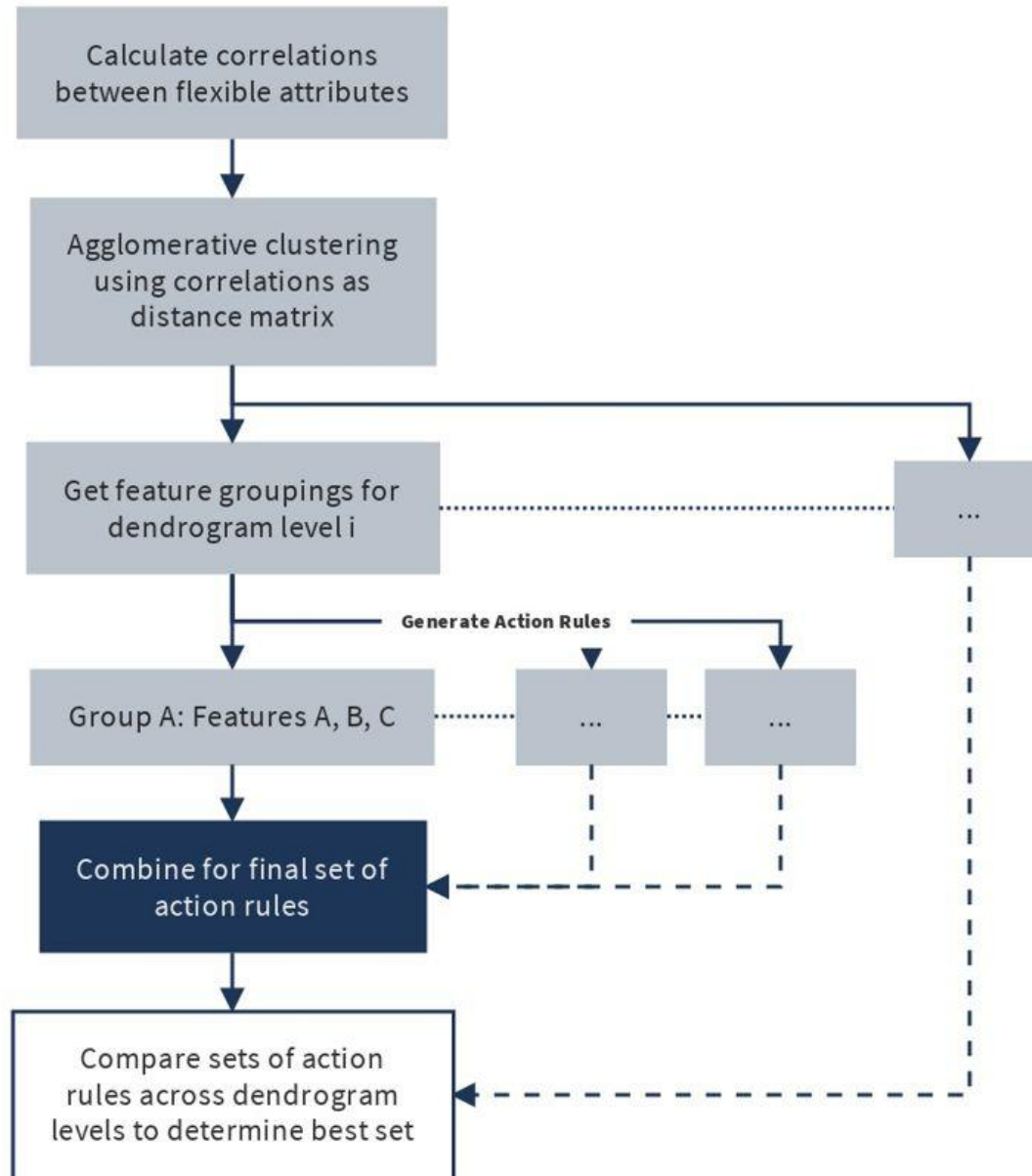
<https://www.ray.io/>



GENERATE ACTION RULES FOR EACH CLUSTER OF FLEXIBLE ATTRIBUTES

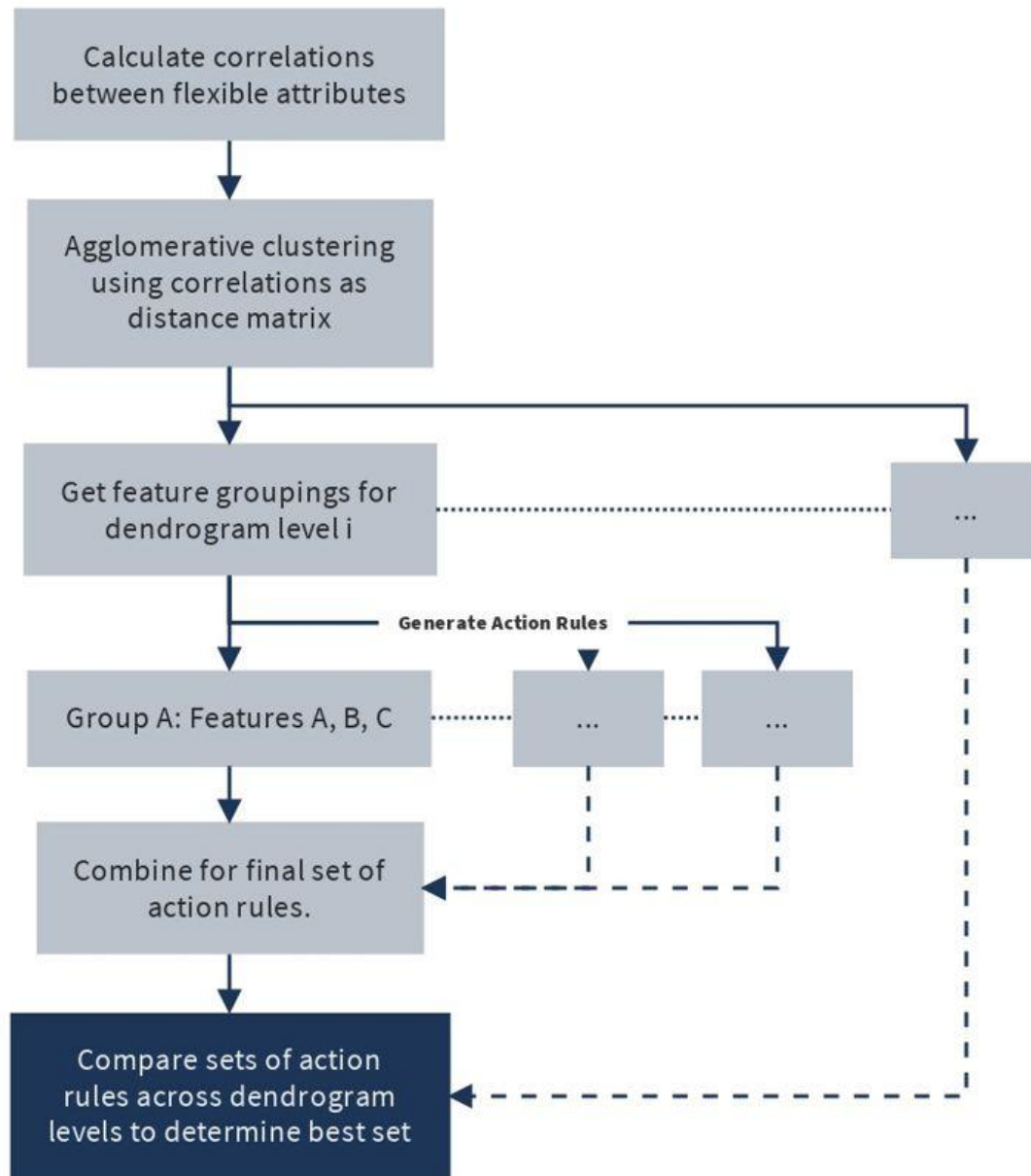
Again, Ray is used here for parallel processing. Action rules are generated from RSES classification rules:

- 1 Initial generation of classification rules from the dataset using RSES (Rough Sets Exploration System).
- 2 Classification rules parsed and structured for further analysis.
- 3 Categorized into bins based on decision attributes, focusing on specific transitions.
- 4 Action rules formulated to recommend attribute value changes for desired transitions.



COMBINE FOR FINAL SET OF ACTION RULES

- Rules concatenated by combining clusters and considering support.
- Enhanced efficiency and precision with depth-controlled approach.
- Confidence and support threshold-based pruning for rule selection.
- Systematic exploration of rule combinations anchored by stable attributes.
- Unique key calculation for streamlined combination process.
- Merging and evaluating rules within each combination.
- Critical assessment against confidence and support thresholds for retention or pruning.



COMPARE SETS OF ACTION RULES ACROSS DENDROGRAM LEVELS TO DETERMINE BEST SET

$$Fscore = 2 \times \frac{(precision \times recall)}{(precision + recall)}$$

We have defined Precision and Recall for a set of rules (R) as:

$$Precision(R) = \frac{\sum_{i=1}^n s_i \times c_i}{\sum_{i=1}^n s_i}, \text{ if } \sum_{i=1}^n s_i > 0$$

REPRESENTING THE WEIGHTED SUM OF CONFIDENCES, WEIGHTED BY THEIR RESPECTIVE SUPPORTS, DIVIDED BY THE TOTAL SUPPORT OF ALL RULES.

$$Recall(R) = \frac{|\bigcup_{i=1}^n I(r_i)|}{N}$$

REPRESENTING THE PROPORTION OF THE TOTAL ITEMS IN THE DATASET THAT ARE COVERED BY THE RULES IN R .

EXPERIMENTAL SETUP



Dataset Background

Net Promoter Score (NPS) Dataset

Decision Attribute: Promoter Status

38 companies and 340,000 customers



Subsets for Experiments

Dataset A: 542 rows, 11 flexible

Dataset B: 1279 rows, 10 flexible

Split by companies.



Data Preprocessing

1. Remove columns based on sparsity threshold
2. Remove redundant columns
3. Handling null values
4. Categorizing benchmarks into bins



Experiment Parameters

Confidence threshold: 80%

Support threshold: 2

Decision attribute (consequent: Passive to Promoter)



Experimental Environment

University Orion Cluster

Each experiment on: one node, 16 CPUs, 4 GB memory per CPU

EVALUATION METRICS

1.Run time

Duration it takes for a method to complete.

2.Number of generated rules

The count of distinct rules produced.

1.Precision

the weighted sum of confidences, weighted by their respective supports, divided by the total support of all rule

2.Coverage

The number of tuples or items encompassed by the rule in the dataset

The coverage for a set of rules is defined by the unique support of all the rules within the set

1.Lightness

A ratio that provides a measure showing how many action rules are discovered that can be applied (on average) to each object in a dataset. More rules means more options to find

The sum of coverages of each action rule divided by the coverage of a set of action rules.

RESULTS AND DISCUSSION

Example of Generated Action Rules

Rule: $[\text{ChannelType} = \text{Construction Heavy}] \wedge$
 $[\text{RC} = [\text{High} \rightarrow \text{Very High}]] \wedge$
 $[\text{OA} = [\text{High} \rightarrow \text{Very High}]]$
 $\Rightarrow [\text{PromoterStatus} = [\text{Passive} \rightarrow \text{Promoter}]]$

Rule Summary

- Rule: *"If the channel type is 'construction heavy', and both the RC (likelihood of customer repeat business) and OA (accuracy of order processing) attributes change from 'high' to 'very high,' then the Promoter Status changes from 'Passive' to 'Promoter'."*
- Confidence: 0.932
- Support: 7

RESULTS

Comparison of Method Performance Metrics		
EXECUTION SPEED		
Both distributed methods outperform the base method.		
The <i>Random</i> method shows faster individual runs than <i>Correlation</i> , however, requires multiple iterations for reliable results.		
For example, the cumulative run time for Dataset A across five iterations of the random method was 823.689 total seconds. The total for Dataset B was 173.808 seco		

Datas et	Method	Time (seconds)	Rules	Precisio n	Coverag e	Lightne ss
A	None	2,543.433	312.0	0.939	0.672	28.171
	Correlati on	393.588	129.0	0.995	0.623	12.816
	Random*	164.738	41.0	0.939	0.525	4.923
B	None	49.077	22.0	0.855	0.418	5.099
	Correlati on	39.924	19.0	0.859	0.418	4.324
	Random*	34.762	4.2	0.681	0.280	1.618

*Random method results are averaged over five iterations.

RESULTS

Comparison of Method
Performance Metrics

QUALITY OF RESULTS

The *Correlation* method outperforms *Random* in delivering a larger number of rules with higher precision, coverage, and lightness in single runs.

Datas et	Method	Time (seconds)	Rules	Precisio n	Coverag e	Lightne ss
A	None	2,543.433	312.0	0.939	0.672	28.171
	Correlati on	393.588	129.0	0.995	0.623	12.816
	Random*	164.738	41.0	0.939	0.525	4.923
B	None	49.077	22.0	0.855	0.418	5.099
	Correlati on	39.924	19.0	0.859	0.418	4.324
	Random*	34.762	4.2	0.681	0.280	1.618

*Random method results are averaged over five iterations.

RESULTS

Comparison of Method
Performance Metrics

RELIABILITY

The *Correlation* method is highlighted for its consistency, reducing the need for averaging over extensive iterations.

For example, one iteration of the Random method for Dataset B resulted in no rules. Multiple necessary iterations of the *Random* method leads to increased overall execution time.

Datas et	Method	Time (seconds)	Rules	Precisio n	Coverag e	Lightne ss
A	None	2,543.433	312.0	0.939	0.672	28.171
	Correlati on	393.588	129.0	0.995	0.623	12.816
	Random*	164.738	41.0	0.939	0.525	4.923
B	None	49.077	22.0	0.855	0.418	5.099
	Correlati on	39.924	19.0	0.859	0.418	4.324
	Random*	34.762	4.2	0.681	0.280	1.618

*Random method results are averaged over five iterations.

CONCLUSION



Objective

Evaluate performance metrics of correlation-based vertical partitioning vs. random and base methods.



Random method:

Shorter completion times but requires multiple iterations due to randomness.



Balance between efficiency and effectiveness:

Correlation-based method outperforms base method in speed and provides more comprehensive results than random method.



Trade-offs observed between execution times, consistency, and quality of generated rules.



Correlation-based method:

Initially more time-consuming but offers consistent results and higher quality rules without the need for repeated iterations.



Future research opportunities:

Further refinement of vertical partitioning methods to enhance overall efficiency and optimize action rule mining.