A comparison of causal discovery algorithms across different datasets

15/01/2026

SUPERVISED BY:
V. CHRISTOPHIDES AND E. VAREILLE

PRESENTED BY:
ANIS AFLOU AND MARTIN GERVAIS

team project ↘

Anis AFLOU

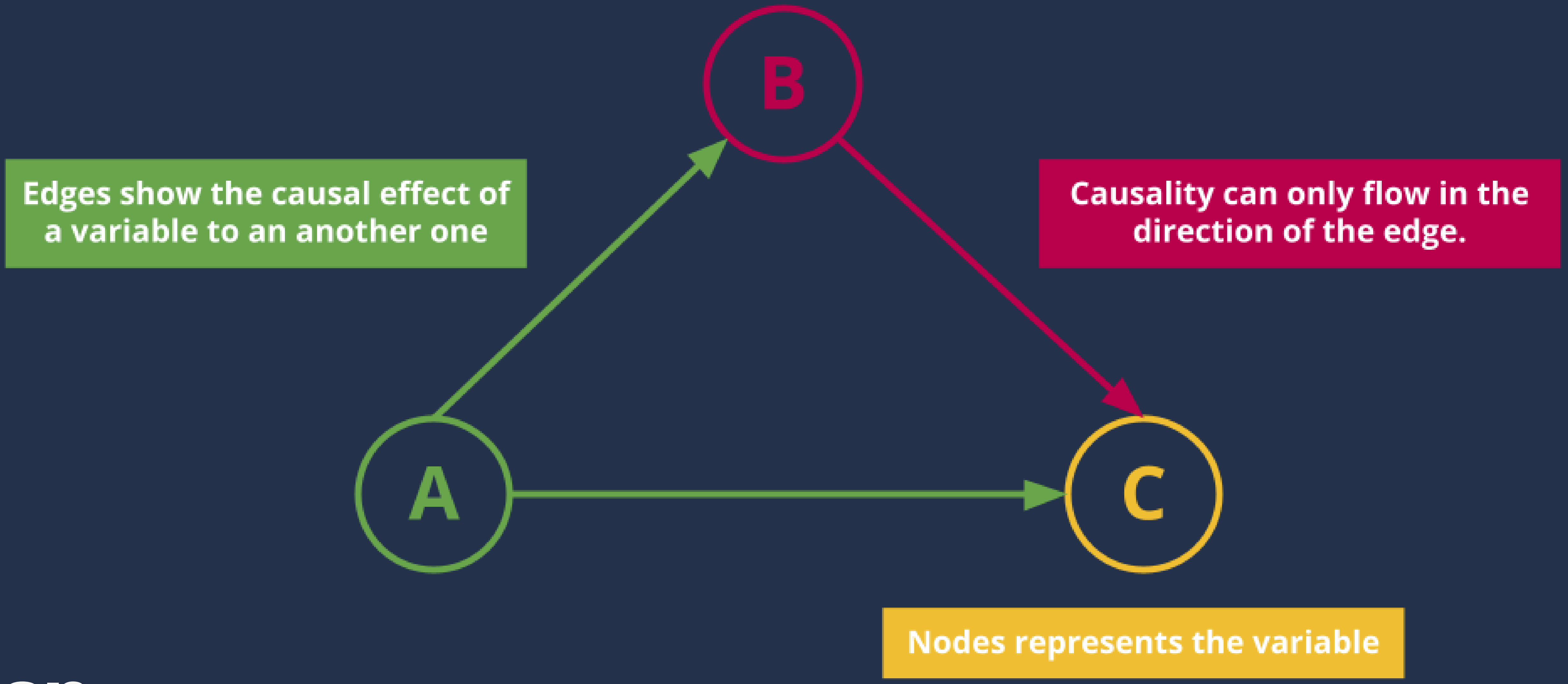
Martin GERVAIS

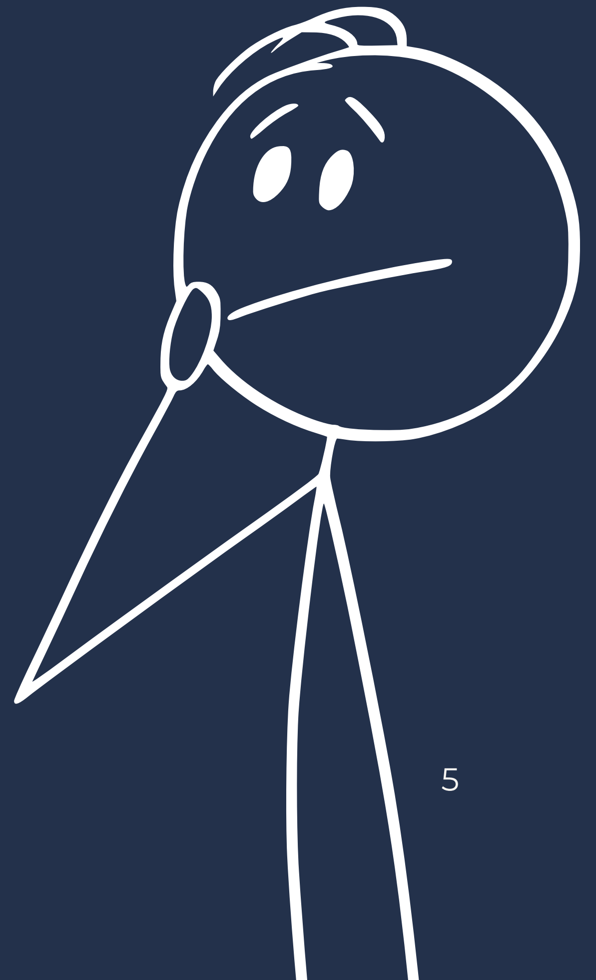
content ↘

context
& motivation

What is **causal discovery**?

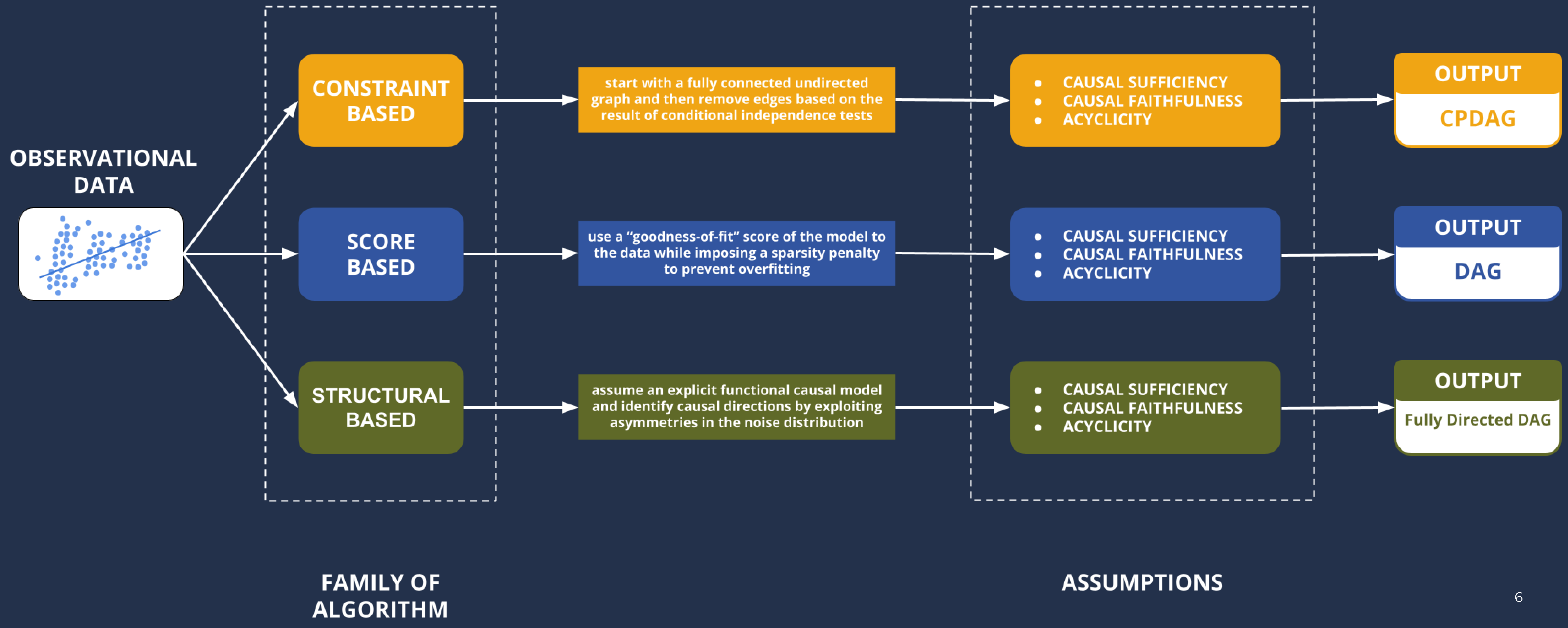
Causal discovery aims to identify causal relationships between variables from the observational data.





context ↗ & motivation

Families of Causal Discovery Algorithms ↘



Why are we comparing different **causal discovery algorithms**?

Many causal discovery algorithms exist, but their practical behavior remains difficult to assess.

1. Numerous causal discovery algorithms have been proposed

2. Algorithms are often tested on different datasets and with different metrics

3. However, evaluations and comparisons are still limited

context & motivation

How do different causal discovery algorithms behave when evaluated under the same experimental conditions?

context ↗
& motivation

project pipeline

project pipeline

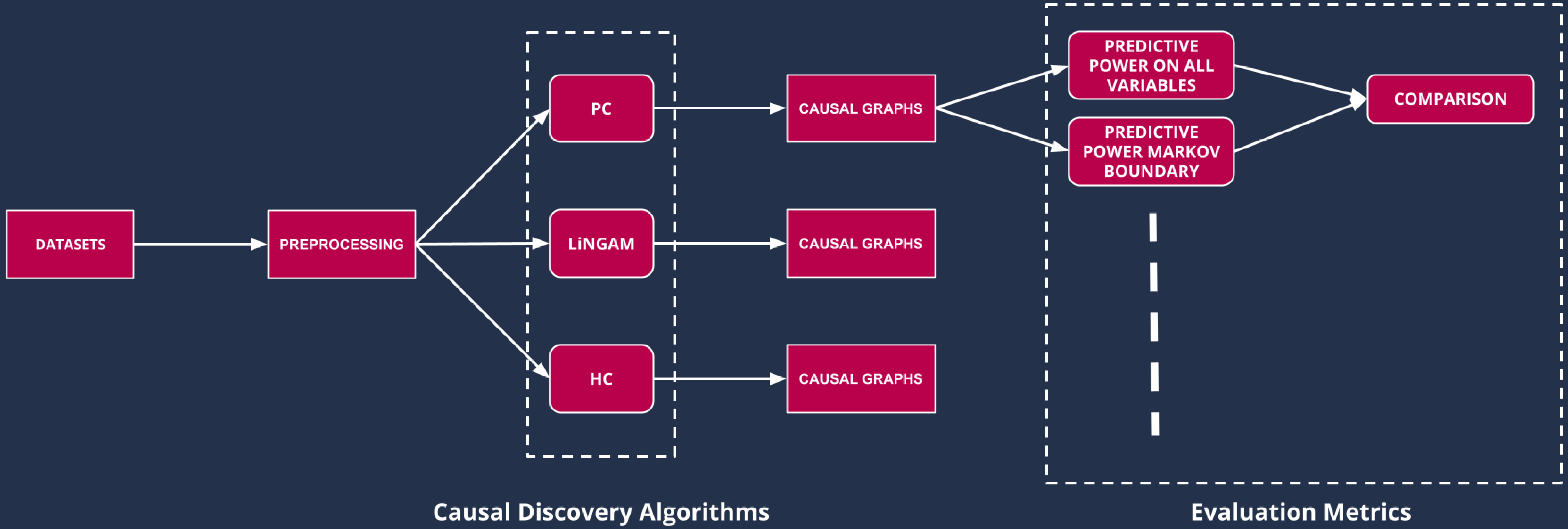


figure 1: experimental pipeline

presentation of the datasets

DATASETS

loan.csv

figure 2: list of the chosen features

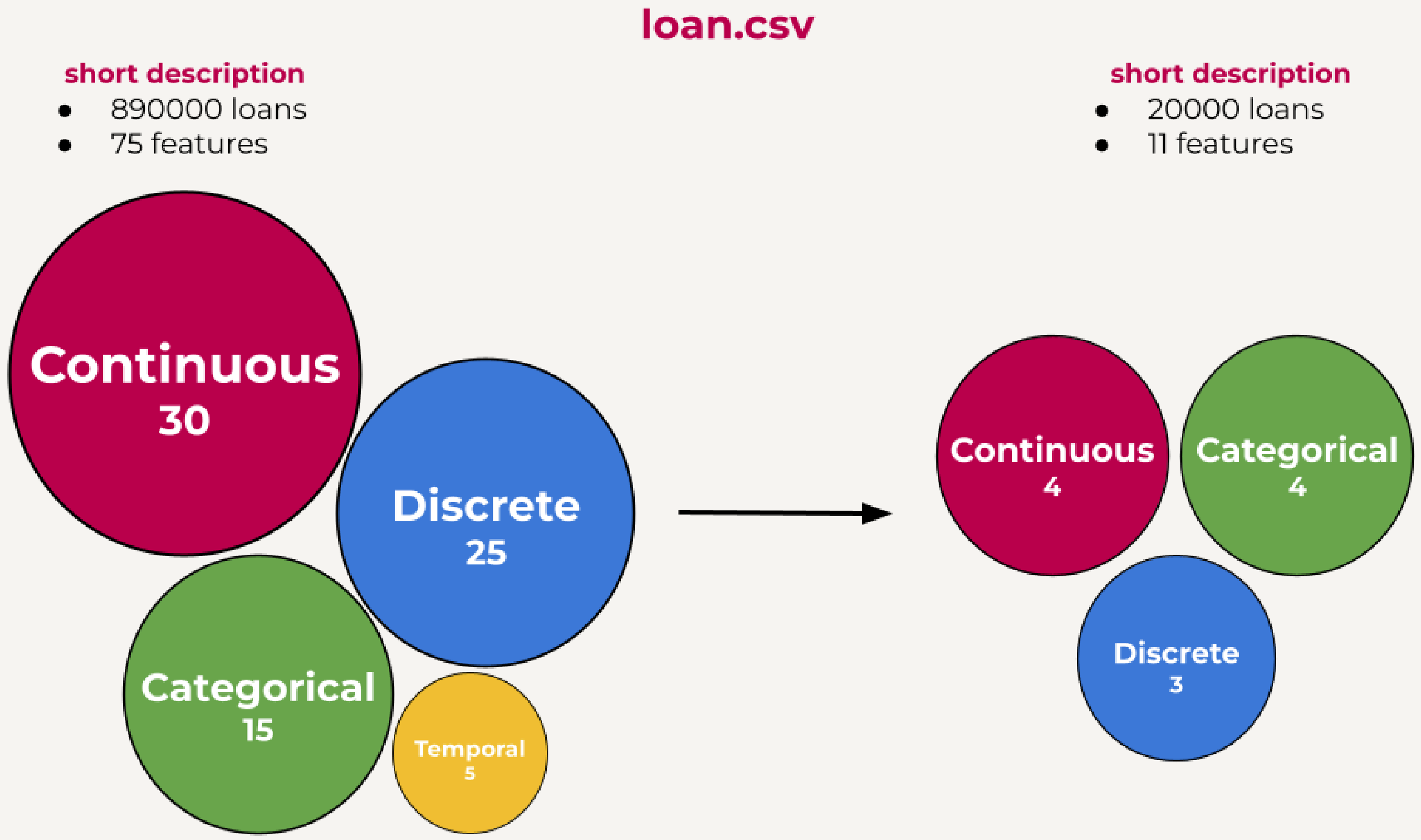


figure 3: bubble charts of loan.csv

presentation ↘
of the datasets

GiveMeSomeCredit.csv

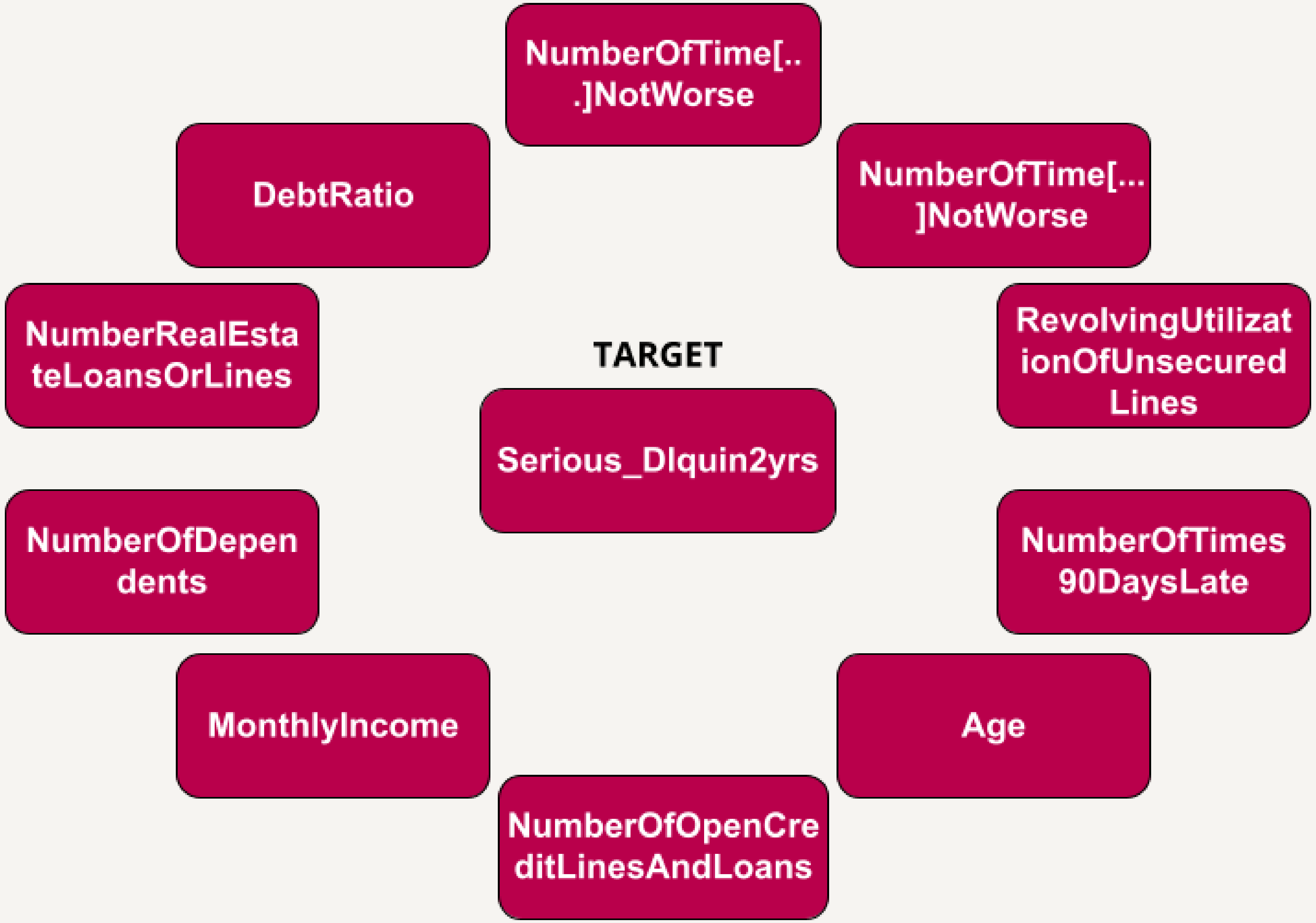


figure 4: list of the chosen features

short description

- 150000 borrowers
- 11 features
- Mixed data

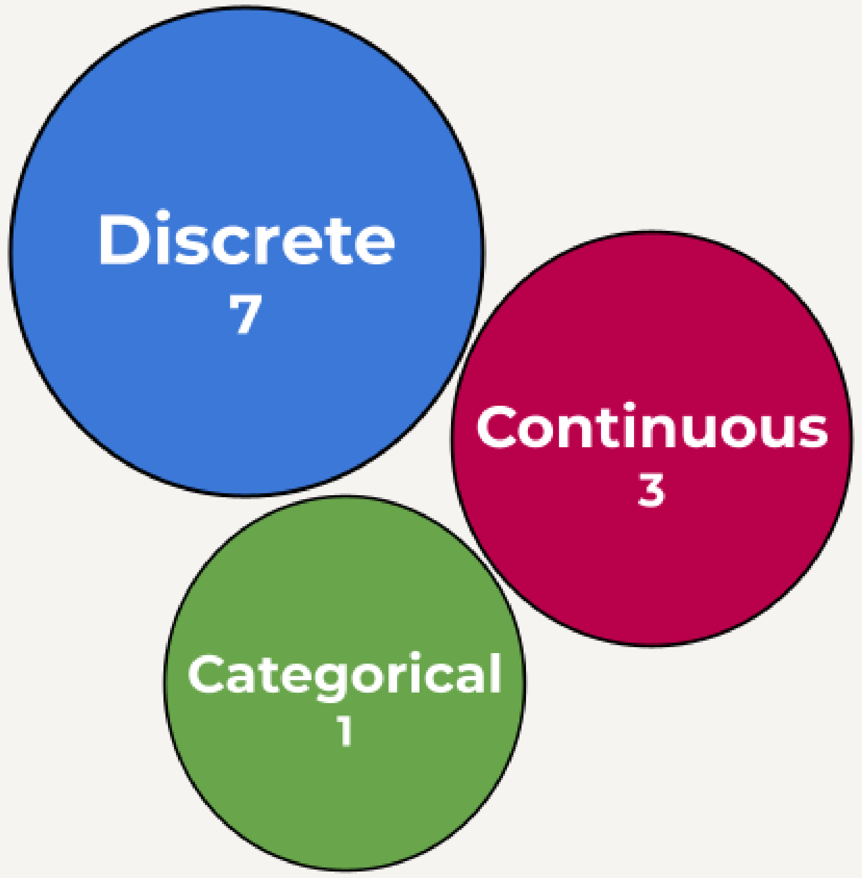


figure 5: bubble chart for GiveMeSomeCredit.csv

OnlineShoppersIntention.csv

figure 6 : list of the chosen features

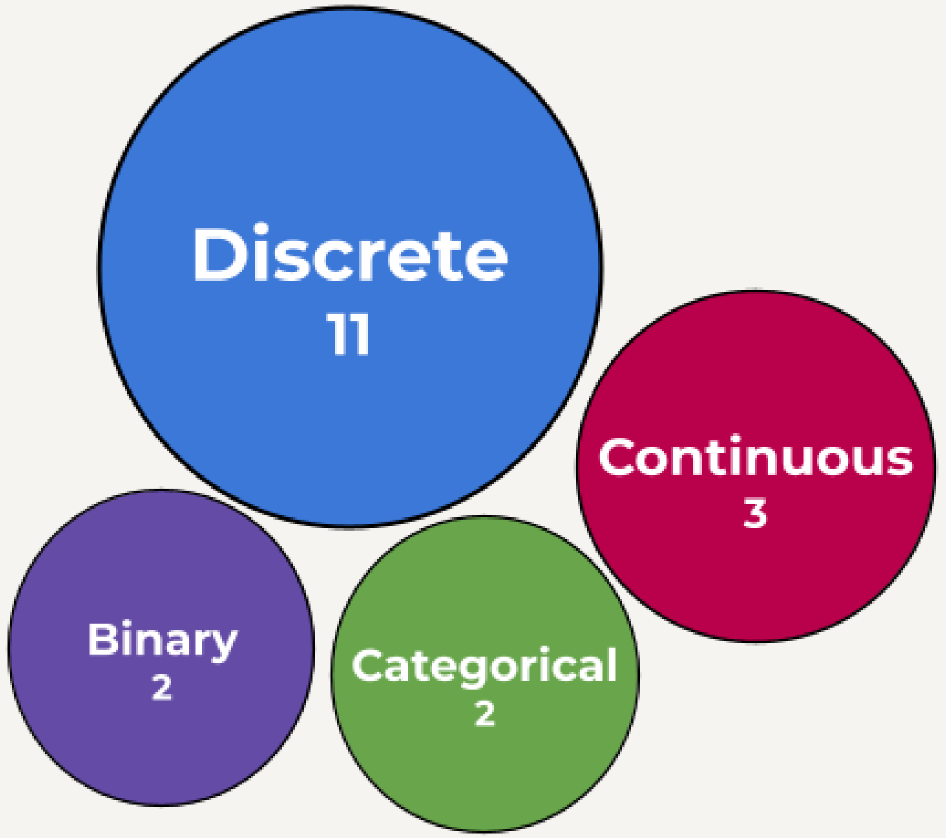


figure 7: bubble chart for OnlineShoppersIntention.csv

short description
- 12330 users sessions
- 18 features

presentation ↘
of the datasets

Algorithms used

PC

LiNGAM

HC

Causal Discovery Algorithms

Algorithms used in this study ↘

	Algorithm	Used on	Assumption	Parameters
01	**PC Algorithm (Constraint Based)**	Mixed datasets (continuous + categorical)	Causal Markov condition (X⊥NonDesc(X) ∣ Parents(X))	ci_test = 'pillai' significance_level=0.05 max_cond _vars = 3 njobs=-1
02	**Hill-Climbing (Score Based)**	Discretized versions of the datasets	Causal Markov condition (X⊥NonDesc(X) ∣ Parents(X))	Scoring method = BIC
03	**LiNGAM (Structural Based)**	Continuous numerical variables	Independent and no gaussian noise	random_state=None, prior_knowledge=None, measure='pwling'

Flowchart of HillClimb Search Model ↘

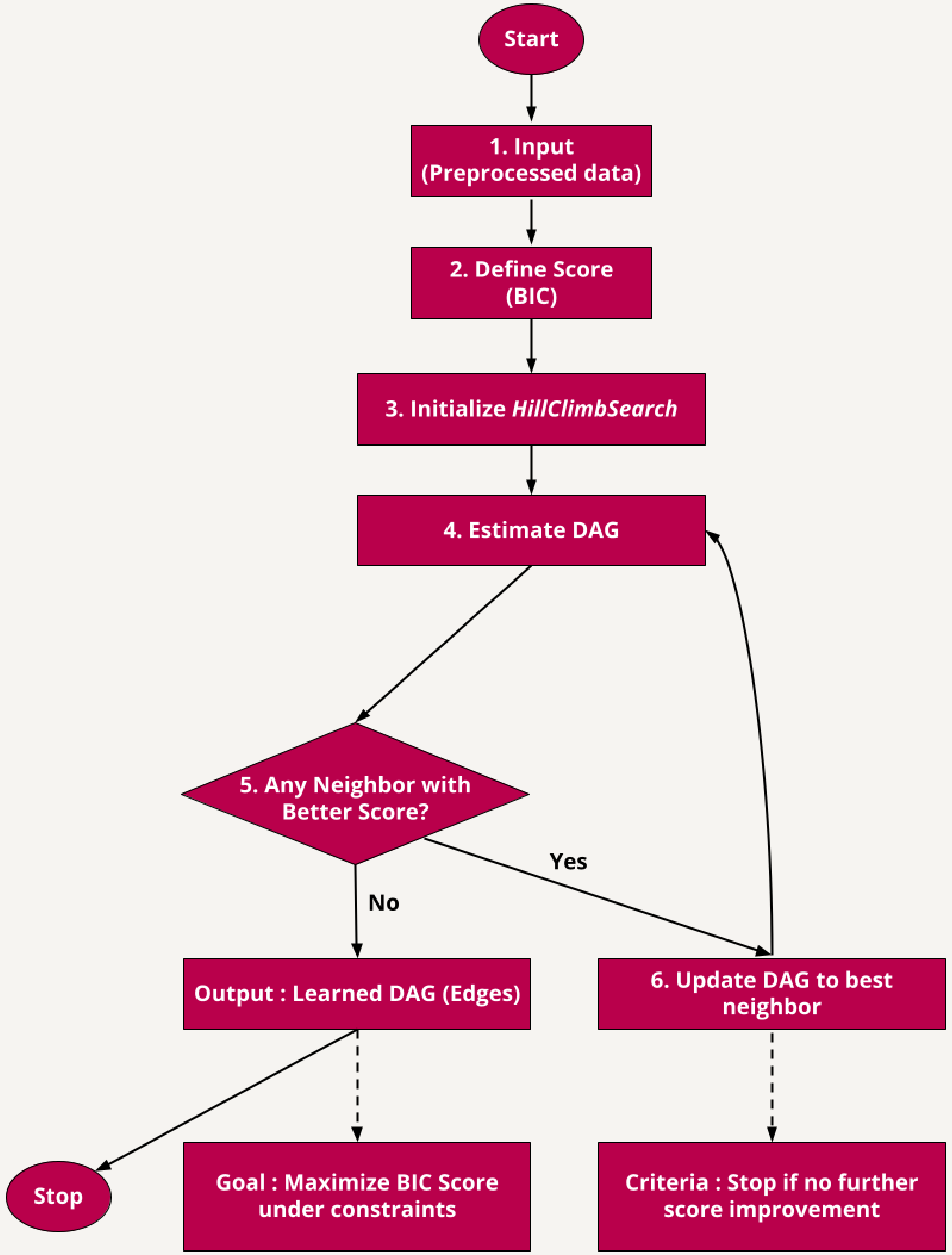


figure 7: flowchart of hillclimb search model

Flowchart of PC Model ↘

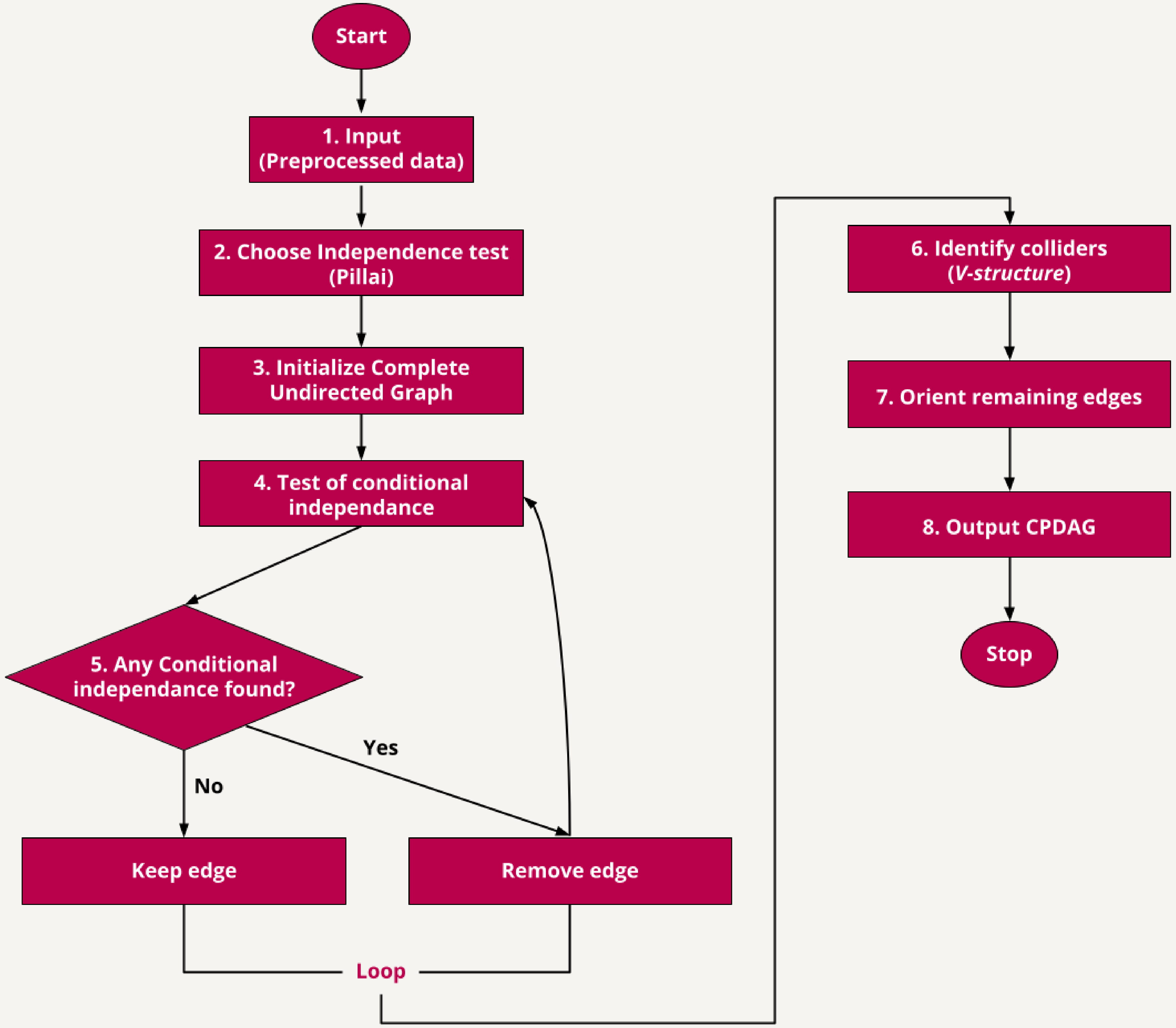


figure 8: flowchart of PC model

Flowchart of LiNGAM Model

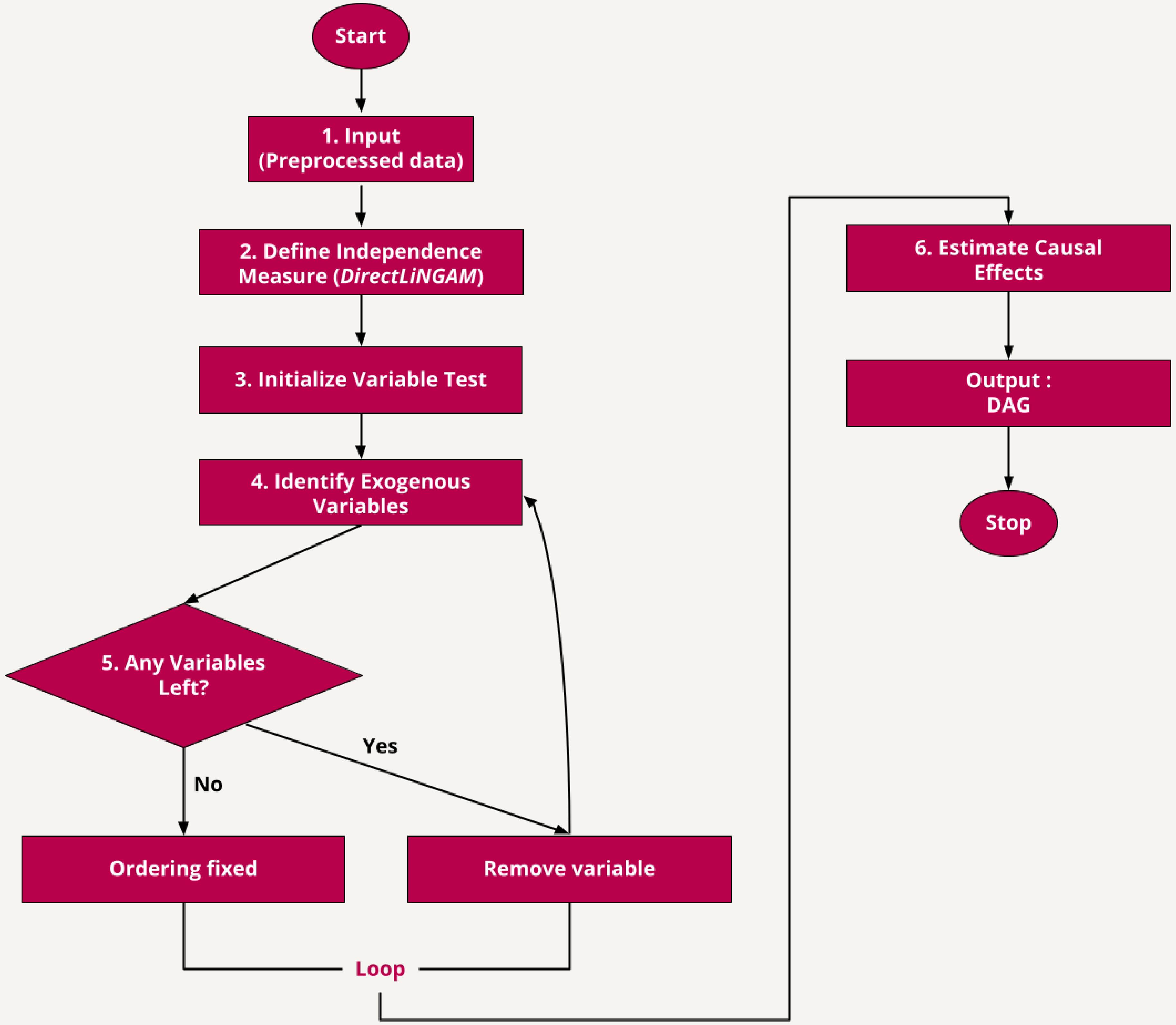
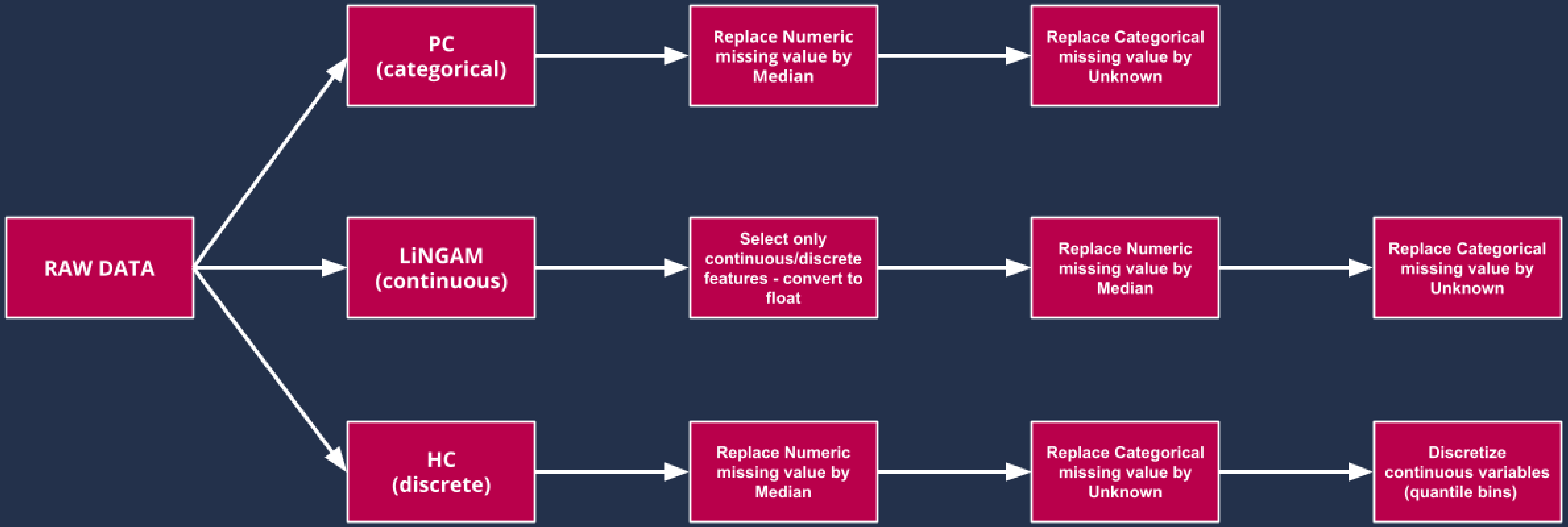


figure 9: flowchart of LiNGAM model

data preprocessing

PREPROCESSING

Preprocessing is adapted to each algorithm's assumptions



Evaluation Metrics ↗

1/ **Predictive Power of Markov Blanket**

- Extract the Markov Blanket of a target variable
- Train a predictive model using MB variables only
- Measure predictive performance

2/ Run time of compilation

***No SHD because we do not have the ground truth**

Evaluation Metrics

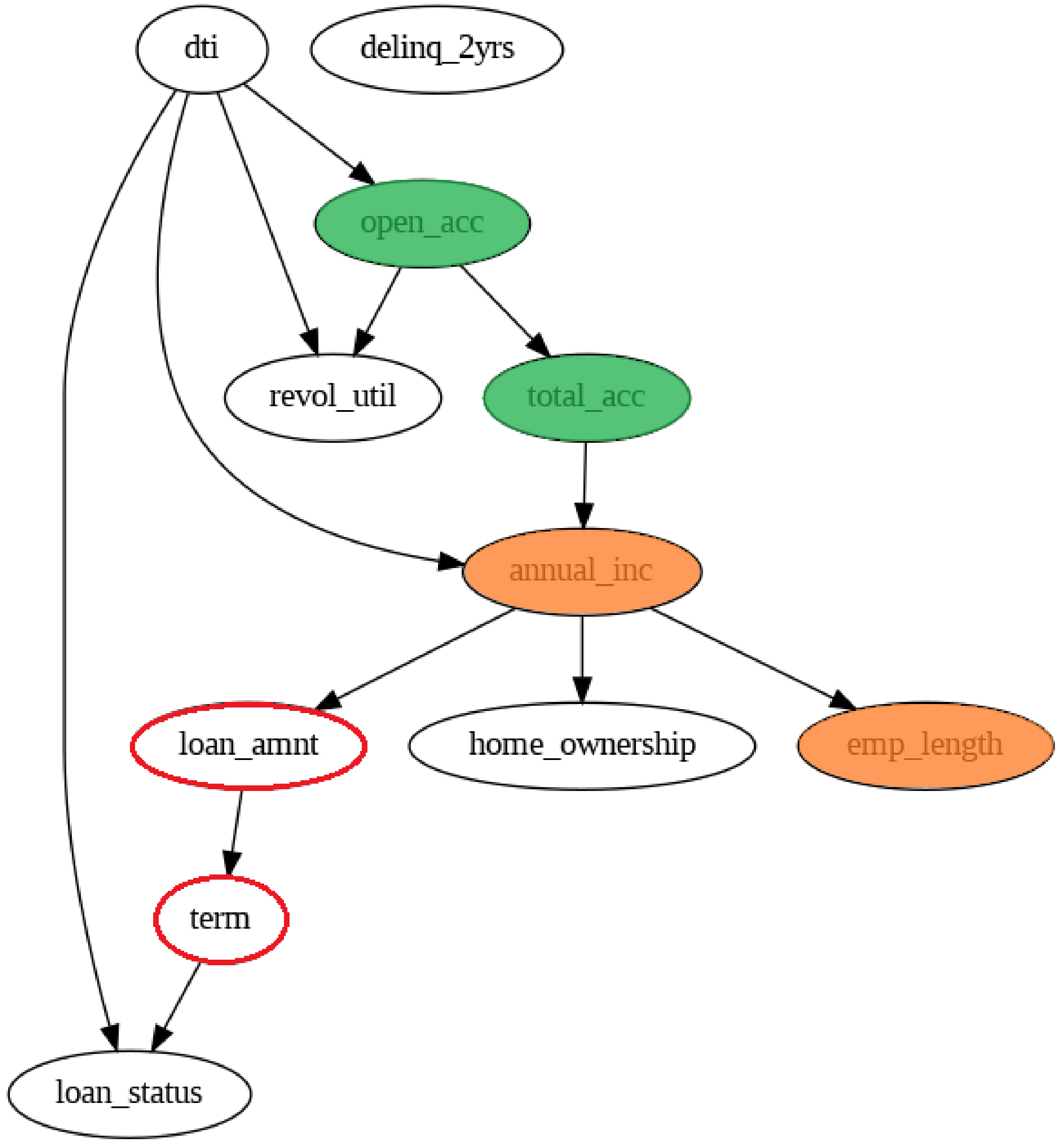
results ↗

1st dataset : loan.csv

loan.csv **PC/HC** comparison

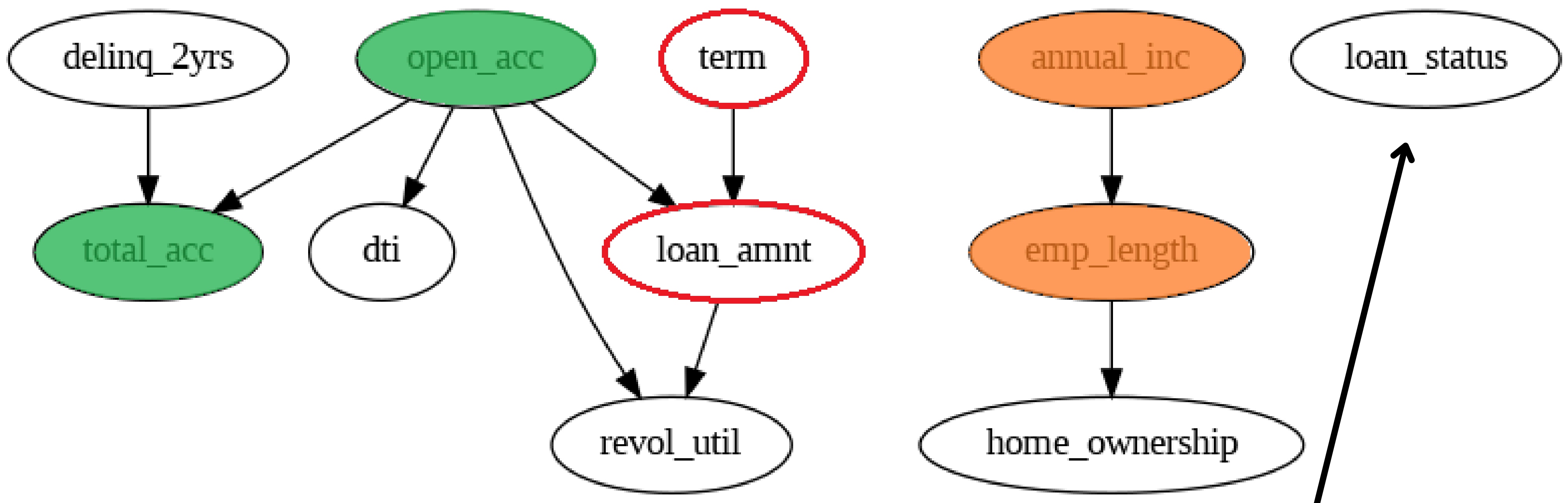
run time → <1s, 8.55it/s

HC algorithm



run time ––> 03:09s, 30.40s/it

PC algorithm



, : same pattern

: same pattern but different orientation

Not many depencies regarding loan_status

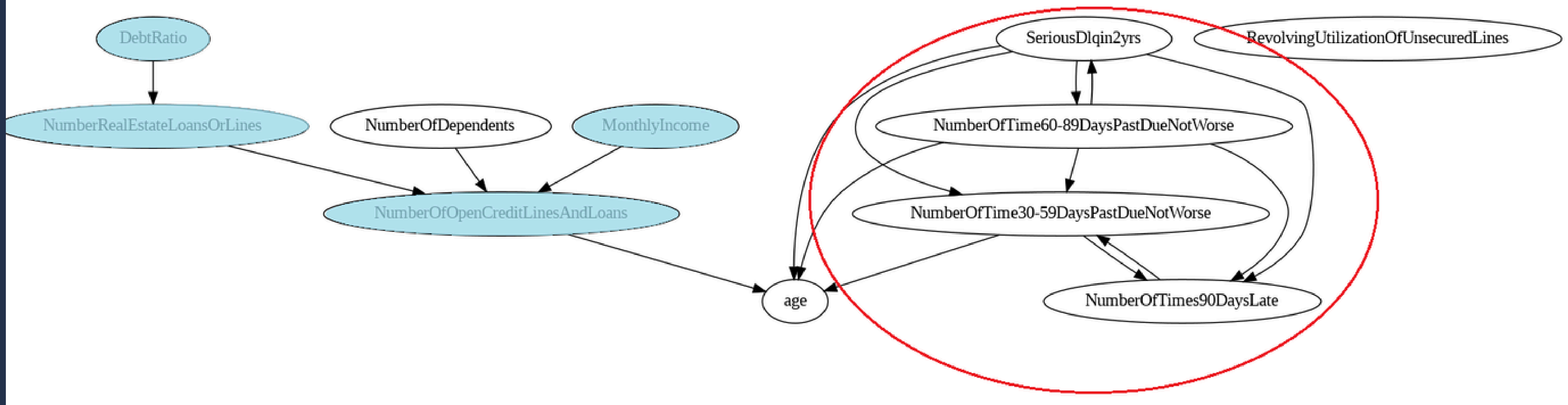
results

figure 10: causal graph on loan.csv with HC algorithm

2nd dataset : GiveMeSomeCredit.csv

GiveMeSomeCredit.csv **PC/HC comparison**

run time → 05:01s, 60.25s/it

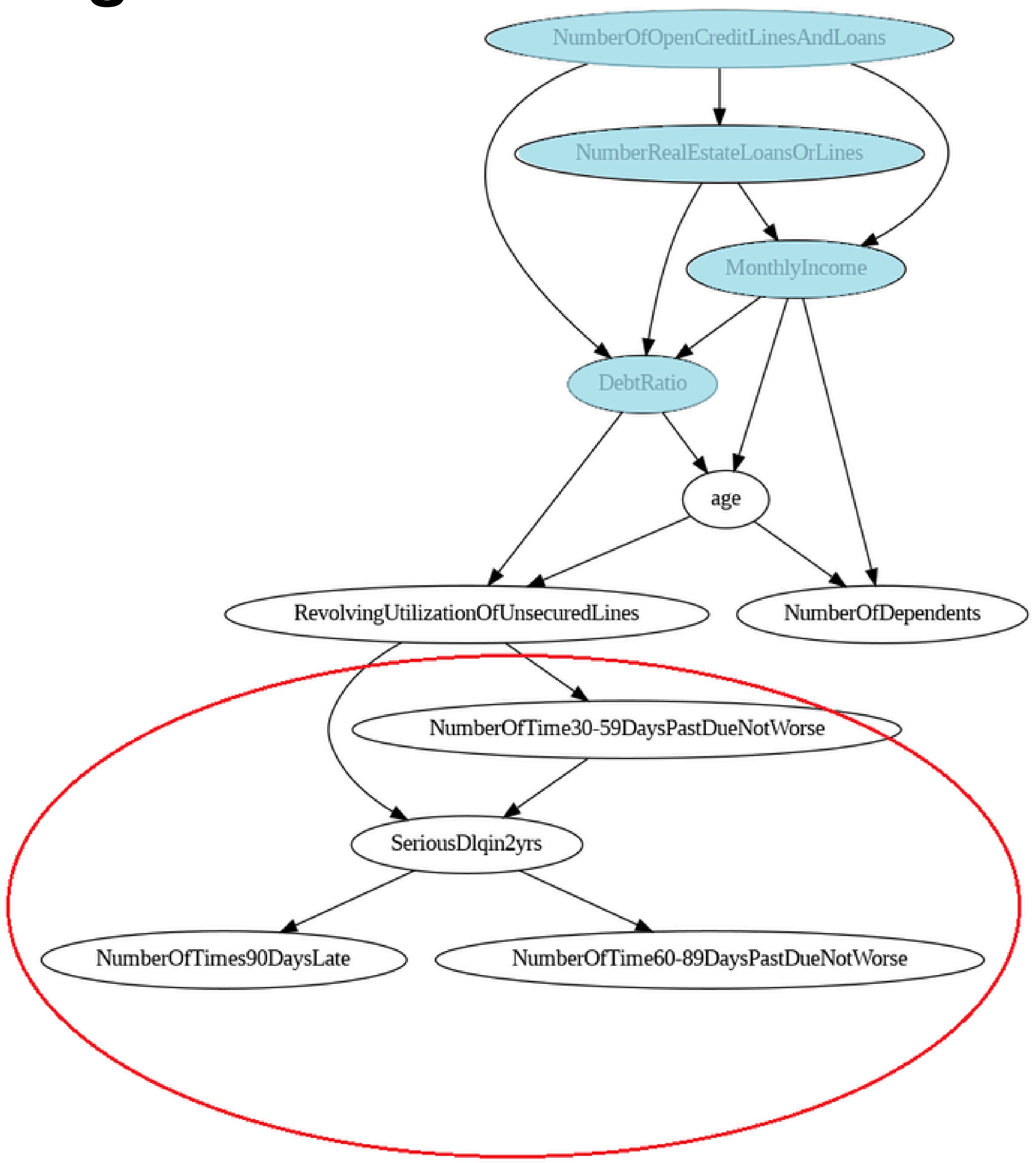


PC algorithm

, : same blocks

run time → <1s, 26.89it/s

HC algorithm



, : same blocks

results

3rd Dataset : OnlineShoppers.csv

OnlineShoppers.csv **PC** algorithm

run time → 30:47s, 397.29s/it

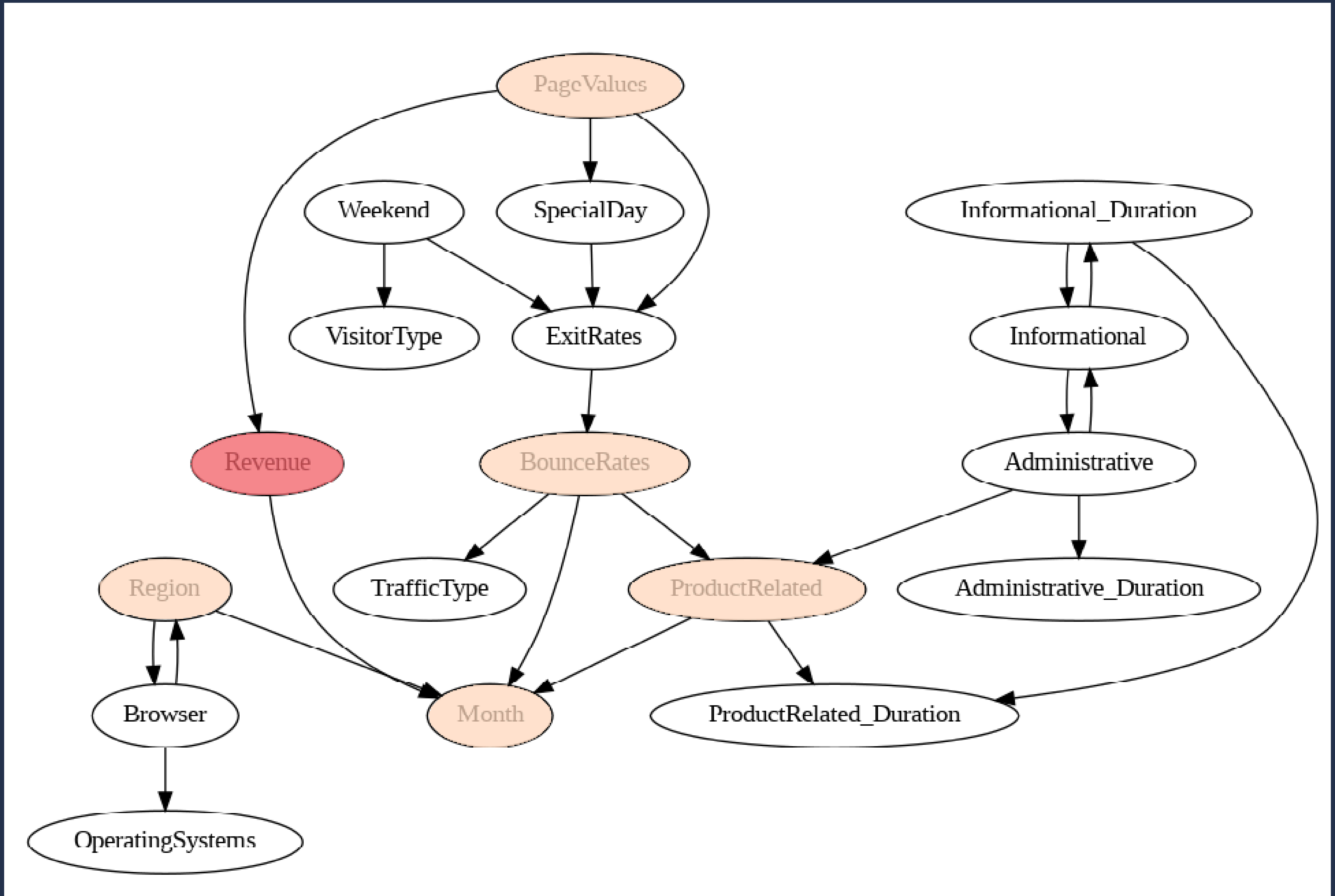


figure 12: causal graph on OnlineShopperscsv with PC algorithm

Markov Blanket

Target Feature

5 features on **Markov Blanket**
[PageValues, BounceRates, ProductRelated, Month, Region]

OnlineShoppers.csv **HC** algorithm

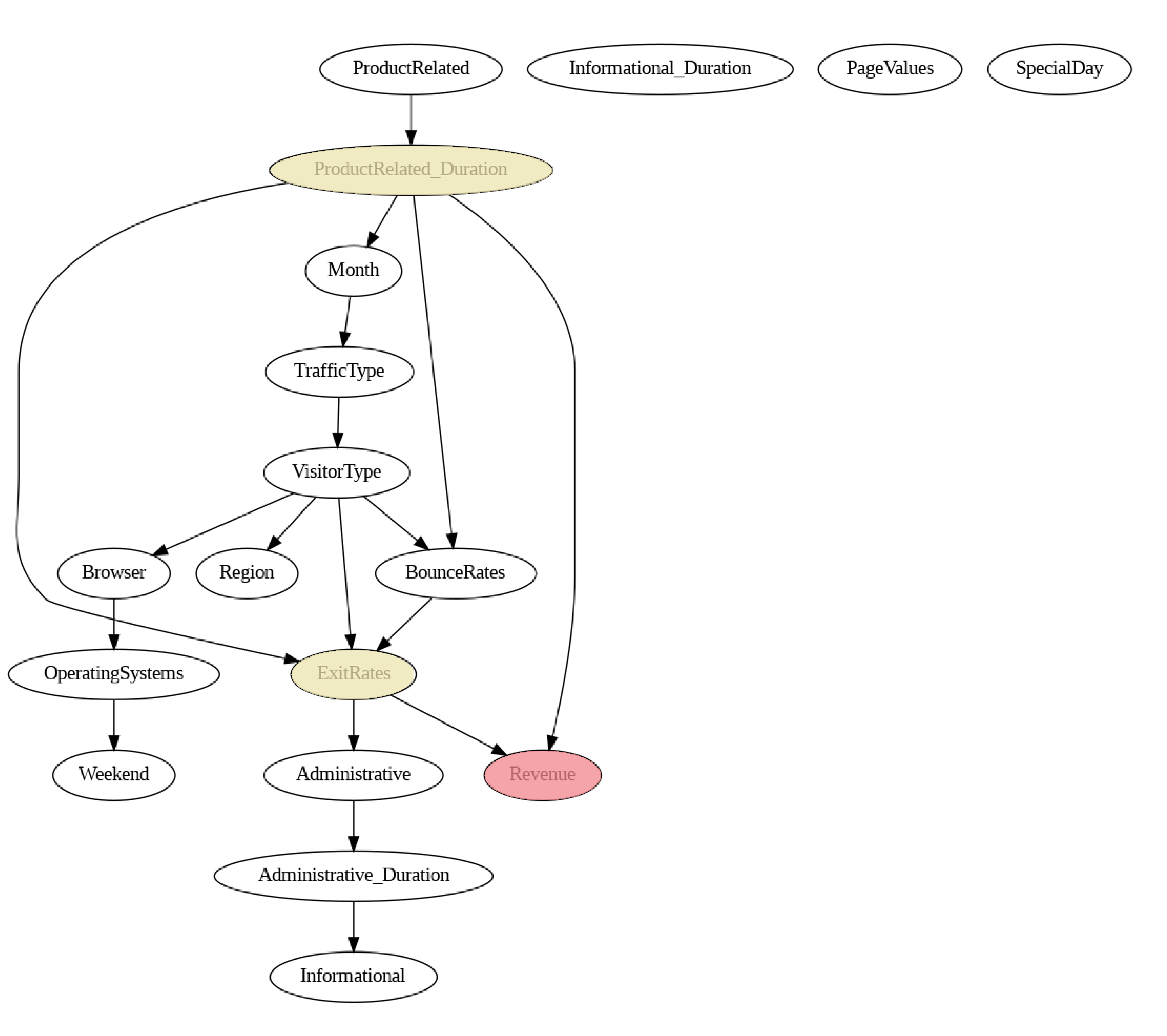


figure 13 causal graph on OnlineShopperscsv with HC algorithm

run time → <1s, 26.21it/s

Markov Blanket

Target Feature

2 features on **Markov Blanket**
(too few)
[ExitRates,
ProductRelated_Duration]

OnlineShoppers.csv **LiNGAM** algorithm

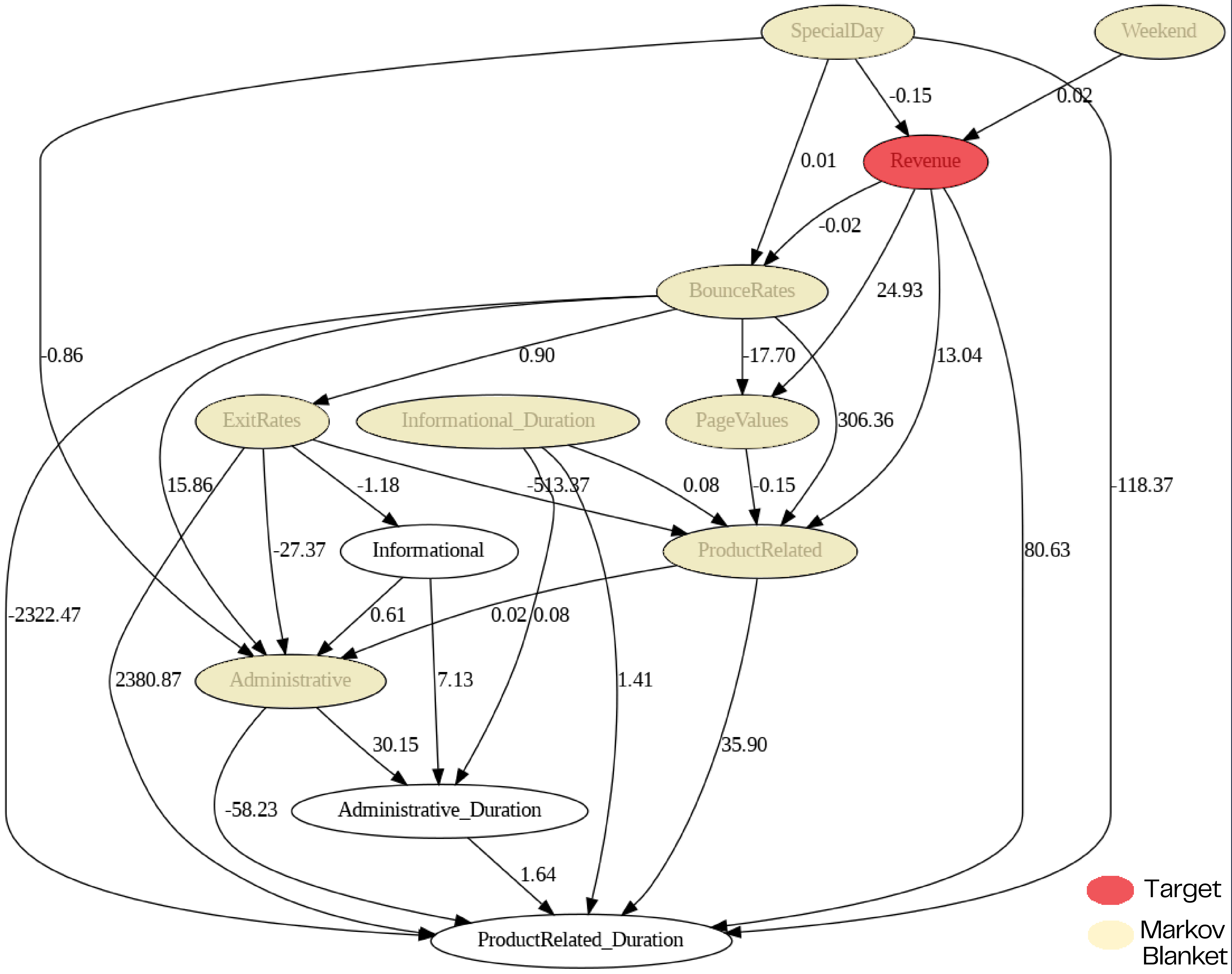


figure 14 causal graph on OnlineShopperscsv with LiNGAM algorithm

run time → 00:15s , 24.11it/s

if X increases of 1 → Y increase by the value

Only on continuous/discrete variables

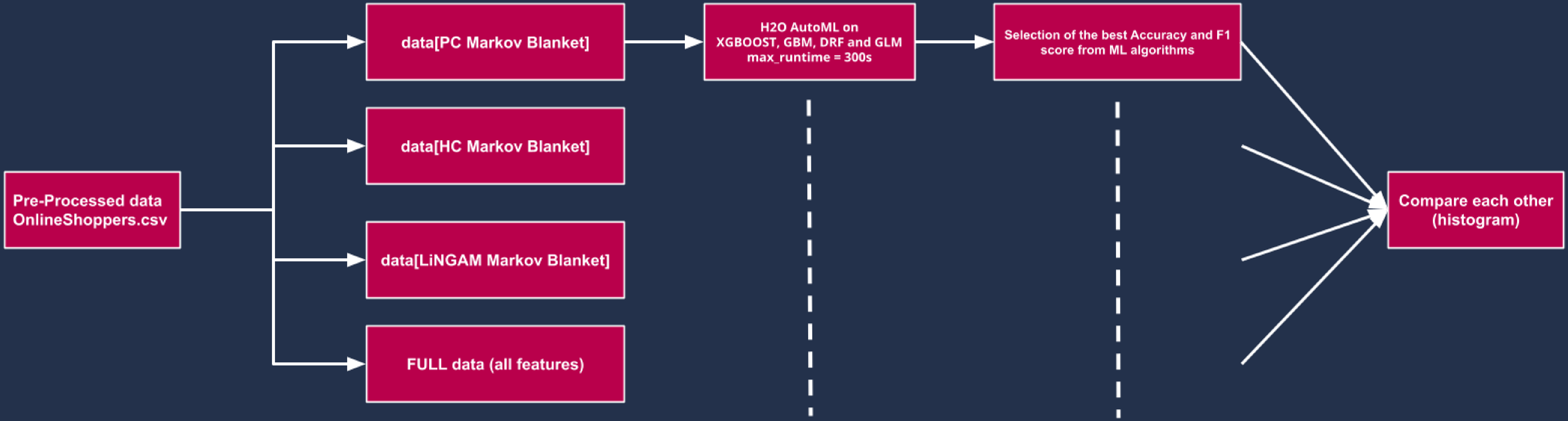
Settings set by default, because of low flexibility with LiNGAM

8 features on **Markov Blanket** [SpecialDay, Weekend, BounceRates, PageValues, ProductRelated, Informational_Duration, ExitRates, Administrative]

Comparaison des algorithmes PC/HC/LiNGAM

Pourquoi ?

- Même condition pour évaluer la puissance prédicitive inter-algorithme
- Gain de temps



Comparaison des algorithmes **PC et HC**

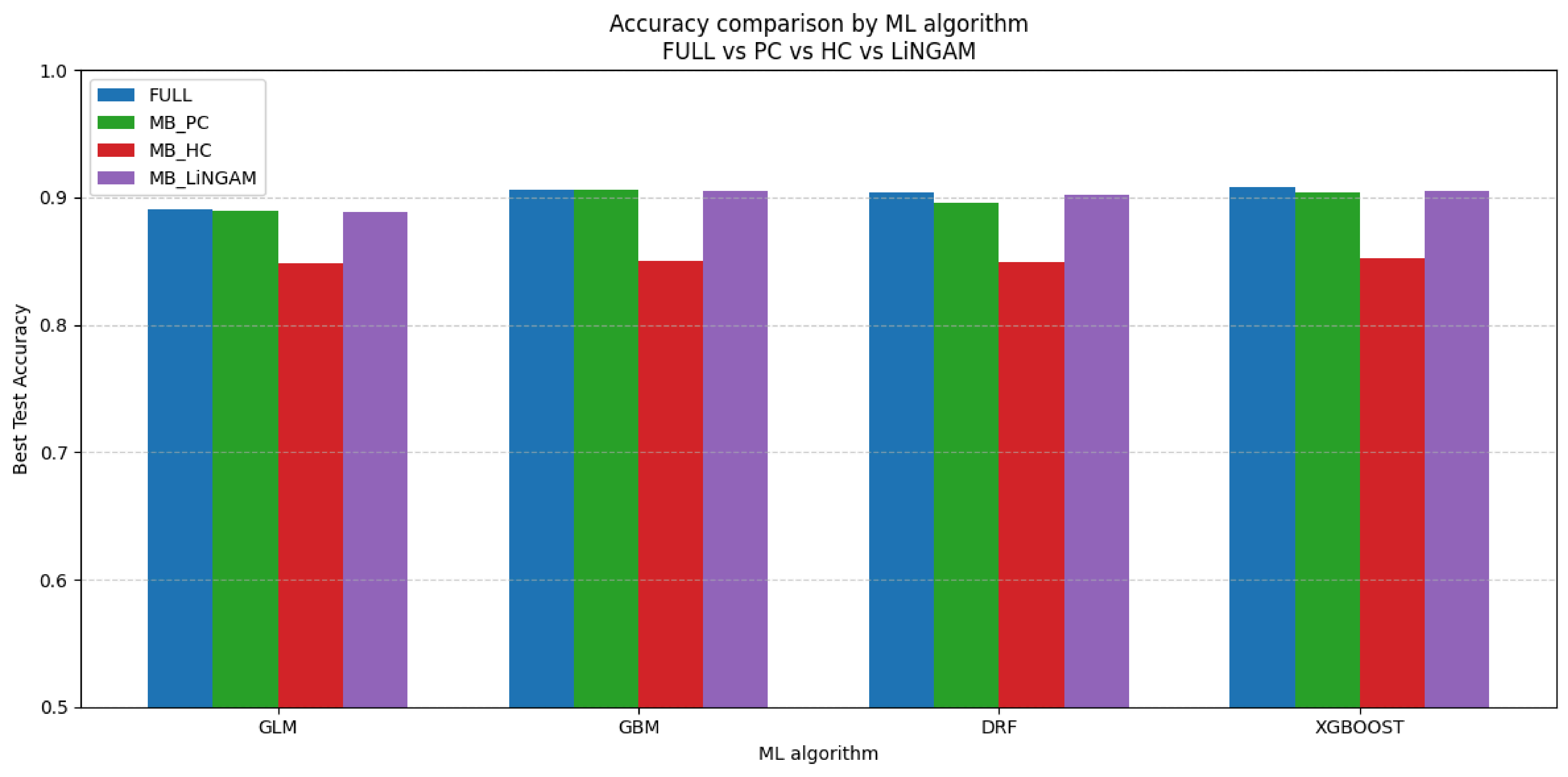


figure 15 : AutoML performance comparison PC/HC

	dataset	Accuracy
0	FULL	0.900990
1	MB_PC	0.897140
2	MB_HC	0.847635
3	MB_LiNGAM	0.903740

LiNGAM donne une meilleur accuracy max avec 8 features contre 18 dans le dataset original !

Comparaison des algorithmes **PC et HC**

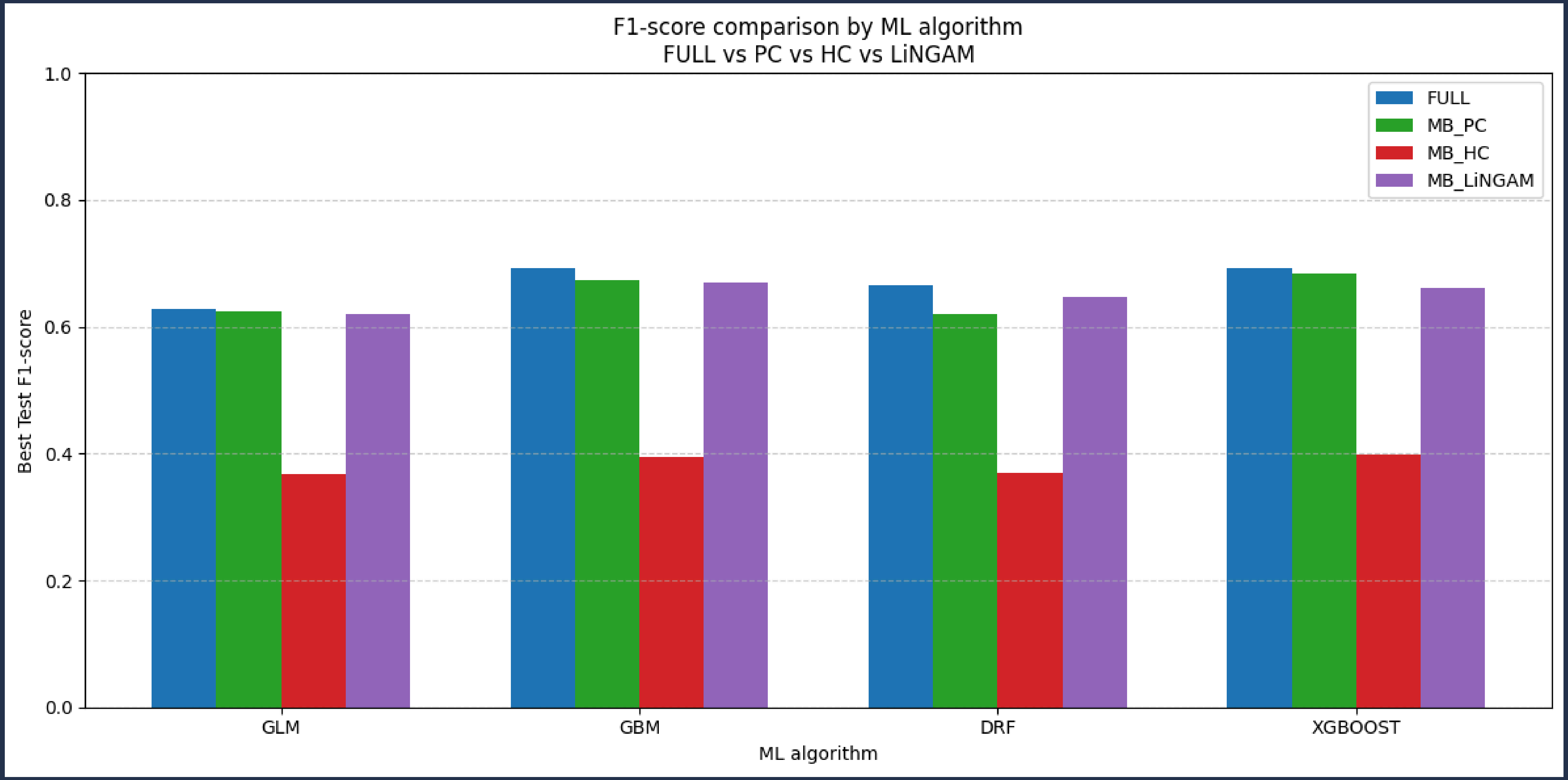


figure 15 : AutoML performance comparison PC/HC

On observe globalement une légère différence entre les F1-Scores du dataset entier par rapport aux autres

	dataset	F1
0	FULL	0.668977
1	MB_PC	0.666667
2	MB_HC	0.380497
3	MB_LiNGAM	0.652241

Petit test par curiosité

MB_UNION = MB_PC U MB_HC

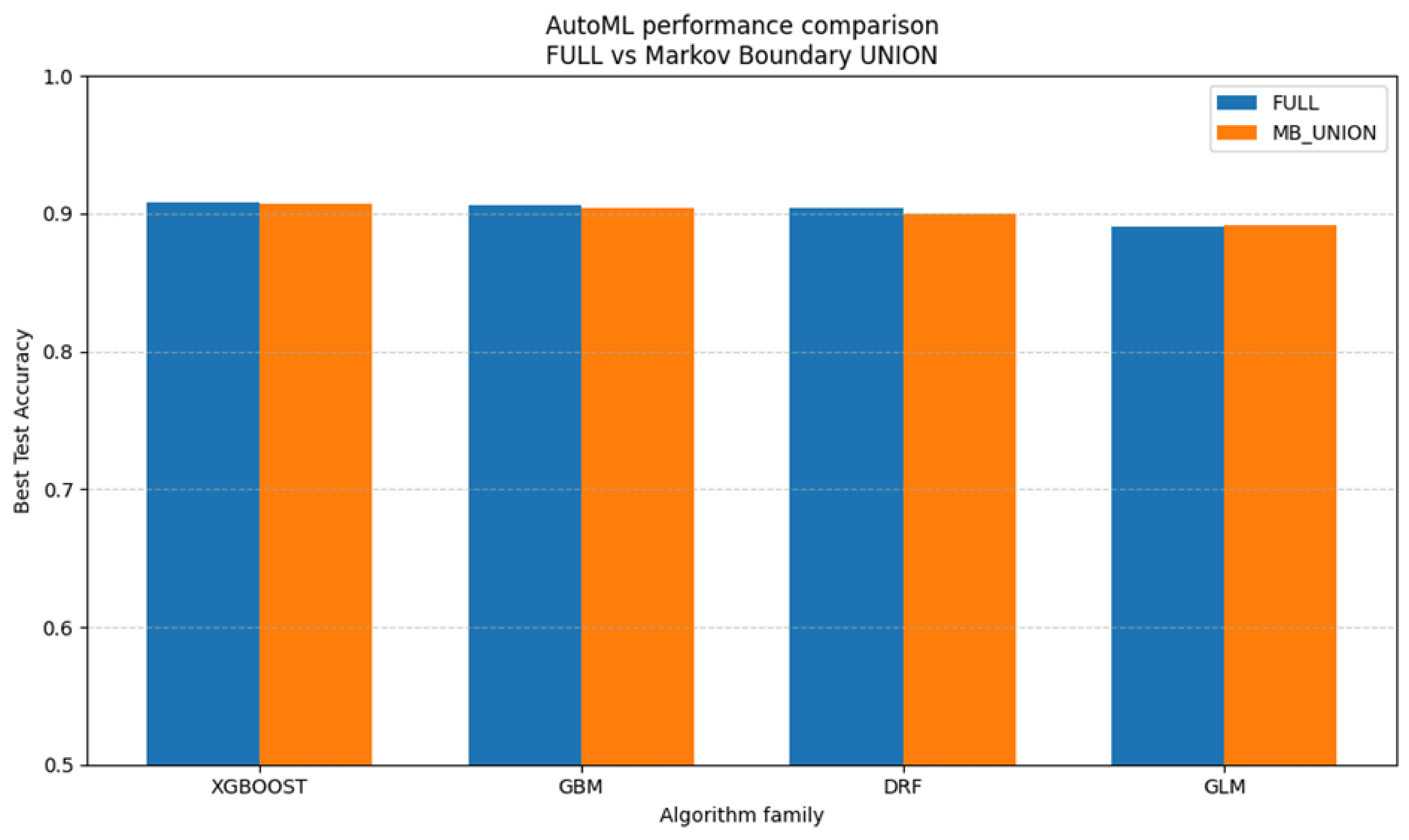


figure 16 : AutoML performance comparison PC U HC

From 18 to 7 features !

Résultat très intéressant

Conclusion

1) Choix des algorithmes de découverte causale

Algorithm	Family	Suitable when	Limitations
PC Peter Clark	Constraint-Based	• Moderate number of data • Mixed data types	• High computational complexity
HC Hill-Climbing	Score-Based	• Trade-off between performance and computation time • Mixed or discretized data	• Possible convergence to local optima
L LiNGAM	Structural-Based	• Mostly continuous variables • Interest in causal effect strengh	• Non Gaussianity required

Perspectives : Découverte causale comme levier de réduction des coût ML

Sur des jeux de données massifs et bruités, l'entraînement de modèles ML devient extrêmement coûteux.

Nous émettons l'hypothèse que :

En identifiant des sous-ensembles (ex. Markov Blanket), on réduirait drastiquement la complexité des modèles ML tout en maintenant, voire en améliorant, leurs performances prédictives.

Cette approche pourrait :

- Réduire le nombre de variables utilisées lors de l'entraînement
- Diminuer les coûts de calcul et de déploiement
- Renforcer l'interprétabilité des modèles

À plus long terme, cela ouvre la voie à des pipelines hybrides causaux-prédictifs, où la découverte causale agit comme un filtre, en amont de modèles ML complexes, combinant ainsi performance et interprétabilité.

Thank you for your attention! ↘