



Faculty of Computer Science  
Data Science Master Program

Ordered Sets for Data  
Analysis

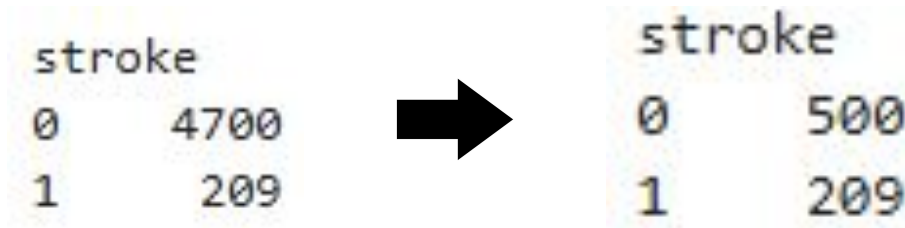
Moscow 2024

# Big HW: Neural FCA

Nasykhova Anastasia



# DATASET



	age	hypertension	heart_disease	avg_glucose_level	bmi	stroke
age	1.000000	0.303512	0.294454	0.316009	0.321822	0.540588
hypertension	0.303512	1.000000	0.123580	0.257510	0.131474	0.294552
heart_disease	0.294454	0.123580	1.000000	0.223900	0.072813	0.228139
avg_glucose_level	0.316009	0.257510	0.223900	1.000000	0.288076	0.263868
bmi	0.321822	0.131474	0.072813	0.288076	1.000000	0.113405
stroke	0.540588	0.294552	0.228139	0.263868	0.113405	1.000000

*Gender*: "Male", "Female" or "Other"

**Age**: The age of the patient.

*Hypertension*: Whether the patient has hypertension (1 for yes, 0 for no).

*Heart\_disease*: Whether the patient has a history of heart disease (1 for yes, 0 for no).

*Ever\_married*: Whether the patient has been married (1 for yes, 0 for no).

Work\_type: The type of work the patient does (e.g., private, self-employed, government job, children, never work).

Residence\_type: Whether the patient resides in an urban or rural area.

**Avg\_glucose\_level**: The patient's average glucose level in blood.

**Bmi**: Body Mass Index.

Smoking\_status: The smoking habit of the patient (e.g., never smoked, formerly smoked, currently smoking or unknown if the information is unavailable for this patient).

**Stroke**: The target variable indicating whether the patient had a stroke (1 for yes, 0 for no).

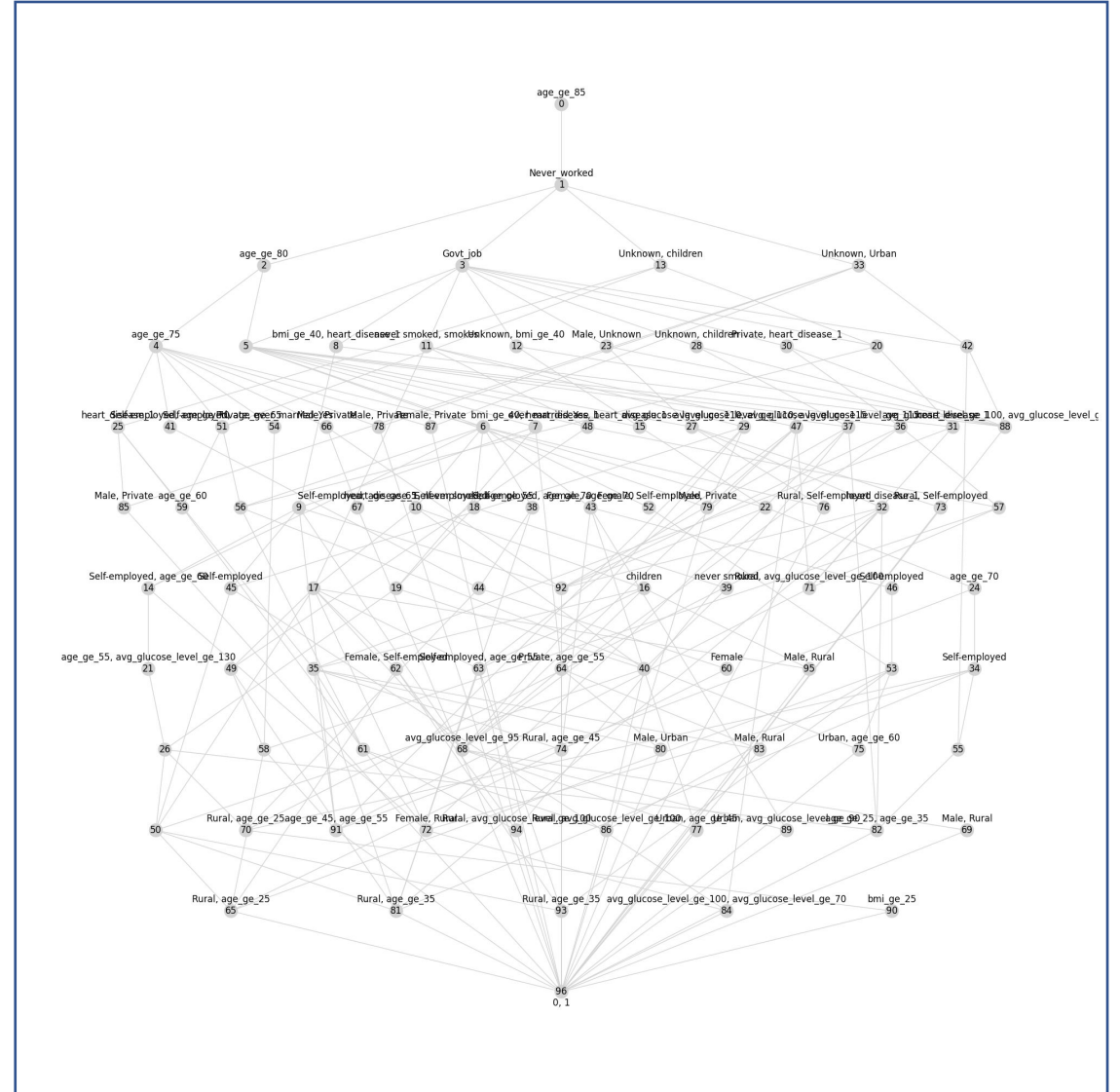
## 1st STRATEGY. BASELINE

```
df_bin_strategy1 = ordinal(
    df_balanced,
    ["age", "avg_glucose_level", "bmi"],
    [
        [25, 35, 45, 55, 60, 65, 70, 75, 80, 85],
        [70, 90, 95, 100, 110, 115, 120, 125, 130, 135],
        [18, 20, 25, 30, 35, 40],
    ],
    style=">=",
)

df_bin_strategy1 = nominal(
    df_bin_strategy1, ["gender", "work_type", "Residence_type", "smoking_status"]
)

df_bin_strategy1 = dichotomic(
    df_bin_strategy1, ["hypertension", "heart_disease", "ever_married"]
)
```

	classifier	accuracy	precision	recall	f1_score
4	Random Forest	0.746479	0.641026	0.531915	0.739809
5	CatBoost	0.739437	0.625000	0.531915	0.733583
6	XGBoost	0.732394	0.609756	0.531915	0.727367
2	Logistic Regression	0.732394	0.615385	0.510638	0.725354
3	Decision Tree	0.718310	0.581395	0.531915	0.714954
1	Naive Bayes	0.704225	0.528090	1.000000	0.707928
0	kNN	0.704225	0.553191	0.553191	0.704225



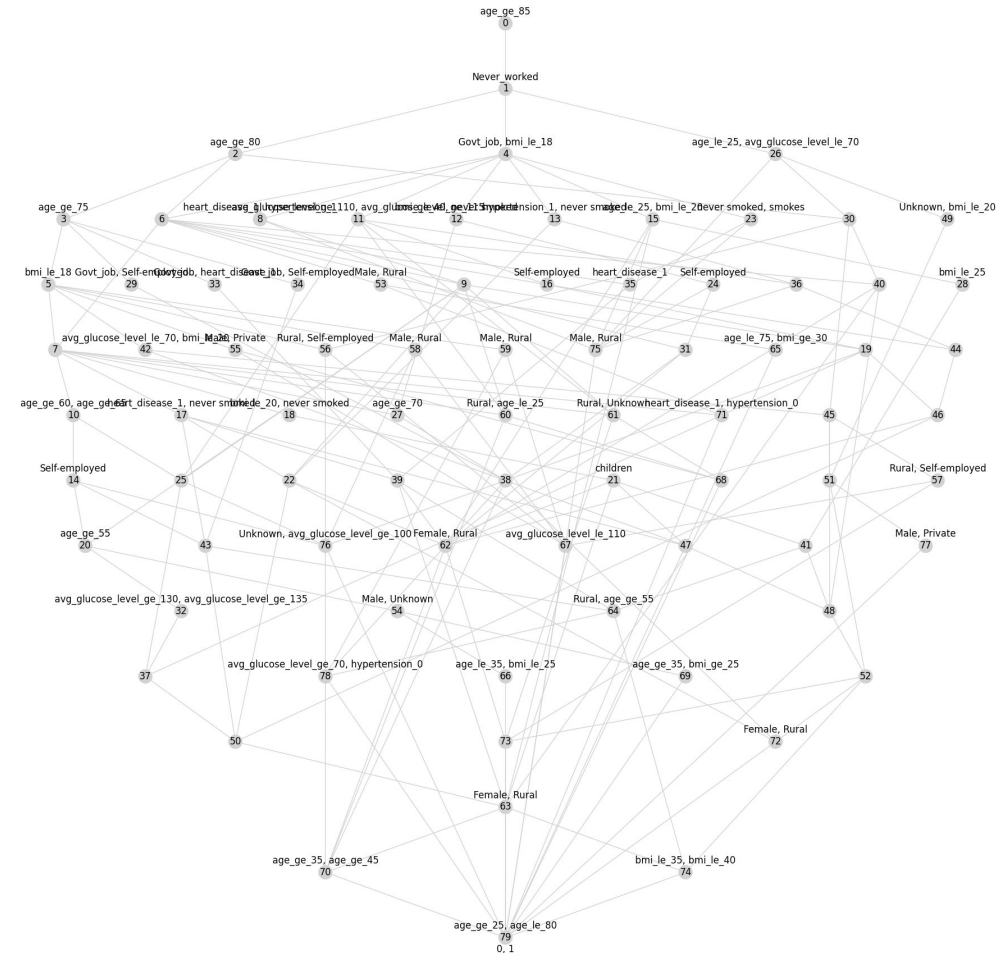
## 2nd STRATEGY. BASELINE

```
df_bin_strategy2 = ordinal(
    df_balanced,
    ["age", "avg_glucose_level", "bmi"],
    [
        [25, 35, 45, 55, 60, 65, 70, 75, 80, 85],
        [70, 90, 95, 100, 110, 115, 120, 125, 130, 135],
        [18, 20, 25, 30, 35, 40],
    ],
    style=">=<=",
)

df_bin_strategy2 = nominal(
    df_bin_strategy2, ["gender", "work_type", "Residence_type", "smoking_status"]
)

df_bin_strategy2 = dichotomic(
    df_bin_strategy2, ["hypertension", "heart_disease", "ever_married"]
)
```

	classifier	accuracy	precision	recall	f1_score
5	CatBoost	0.739437	0.613636	0.574468	0.737167
2	Logistic Regression	0.739437	0.625000	0.531915	0.733583
1	Naive Bayes	0.718310	0.542169	0.957447	0.724388
0	kNN	0.718310	0.574468	0.574468	0.718310
3	Decision Tree	0.718310	0.581395	0.531915	0.714954
4	Random Forest	0.718310	0.589744	0.489362	0.710899
6	XGBoost	0.718310	0.589744	0.489362	0.710899





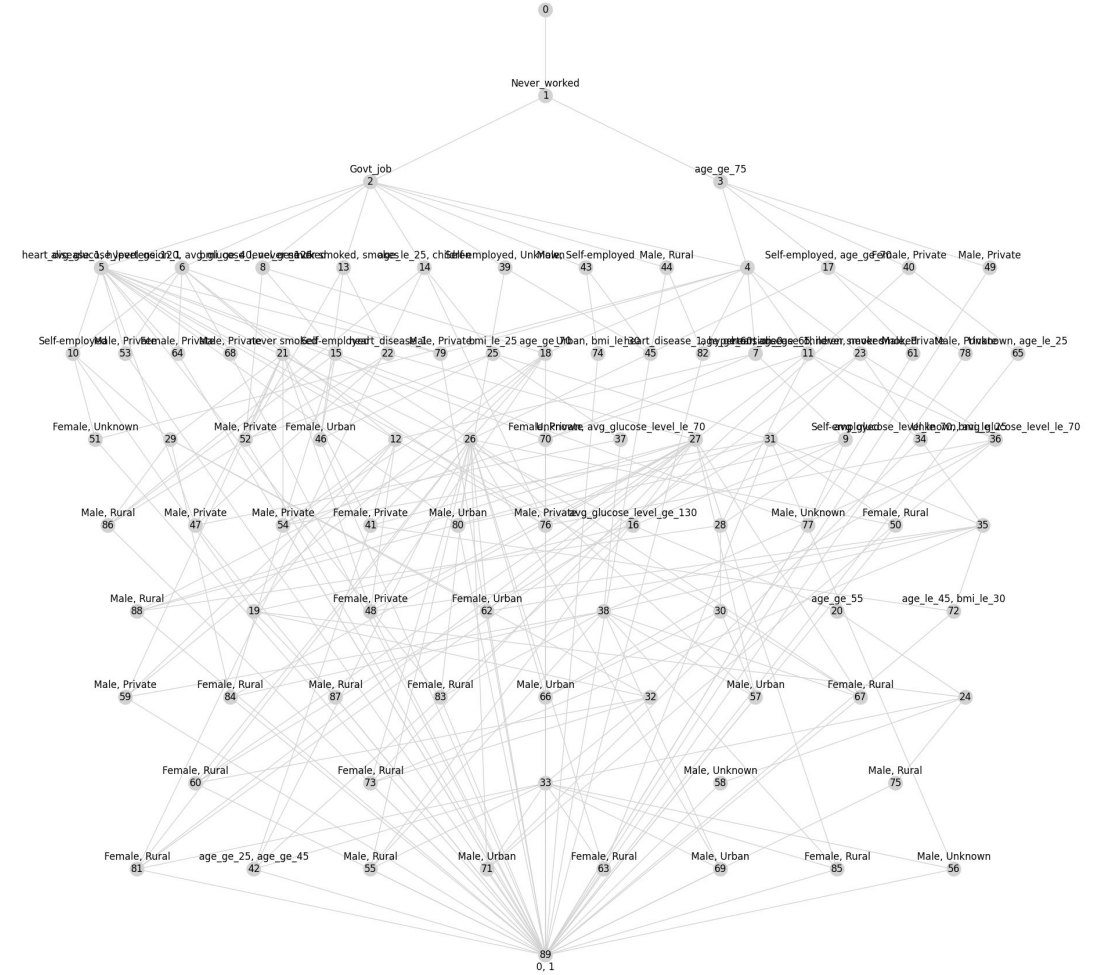
### 3rd STRATEGY. BASELINE

```
df_bin_strategy3 = ordinal(
    df_balanced,
    ["age", "avg_glucose_level", "bmi"],
    [
        [25, 45, 55, 60, 65, 70, 75],
        [70, 90, 95, 120, 125, 130],
        [25, 30, 35, 40],
    ],
    style=">=<=",
)

df_bin_strategy3 = nominal(
    df_bin_strategy3, ["gender", "work_type", "Residence_type", "smoking_status"]
)

df_bin_strategy3 = dichotomic(
    df_bin_strategy3, ["hypertension", "heart_disease", "ever_married"]
)
```

	classifier	accuracy	precision	recall	f1_score
1	Naive Bayes	0.767606	0.592105	0.957447	0.774072
4	Random Forest	0.746479	0.634146	0.553191	0.741717
5	CatBoost	0.739437	0.613636	0.574468	0.737167
2	Logistic Regression	0.739437	0.625000	0.531915	0.733583
6	XGBoost	0.732394	0.609756	0.531915	0.727367
3	Decision Tree	0.725352	0.611111	0.468085	0.714668
0	kNN	0.704225	0.558140	0.510638	0.700702

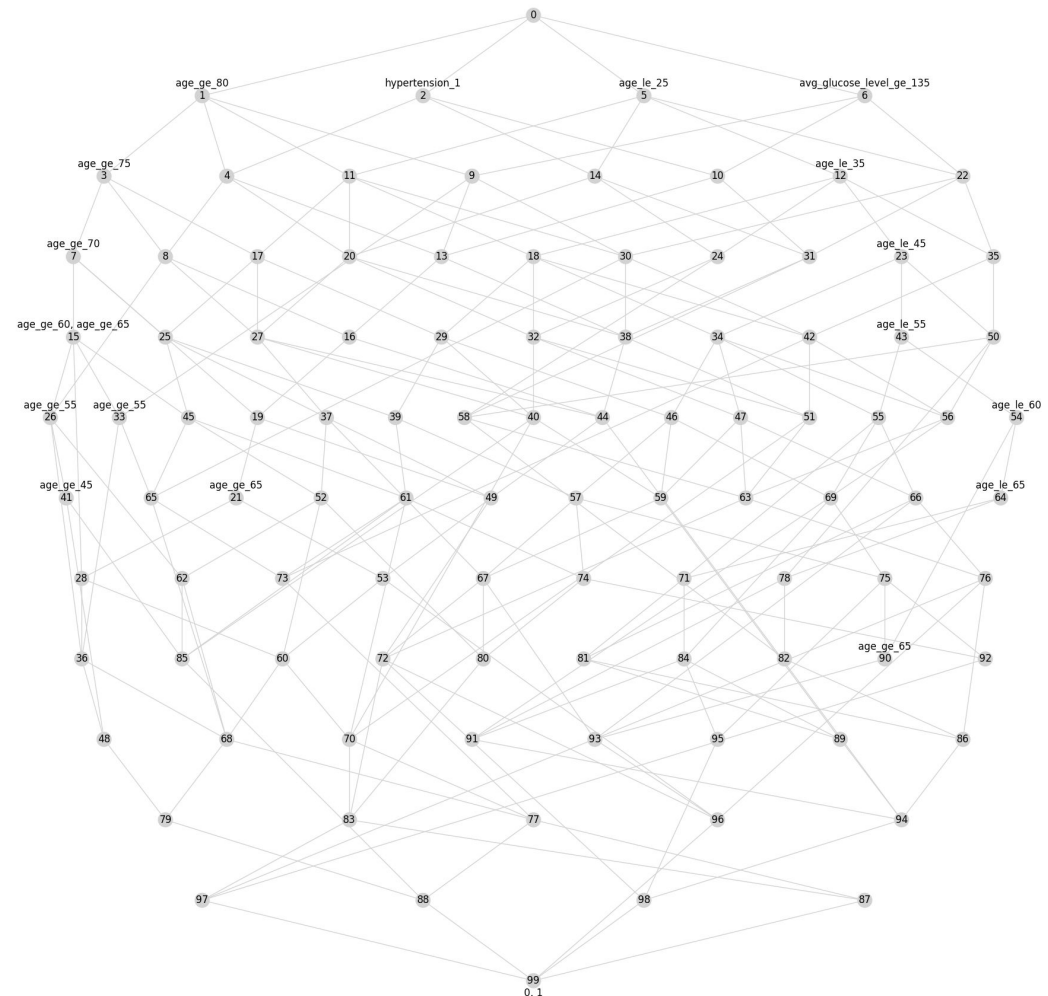




## 4th STRATEGY. BASELINE

feature with column\_index 1 = age\_le\_25  
feature with column\_index 3 = age\_le\_35  
feature with column\_index 4 = age\_ge\_45  
feature with column\_index 5 = age\_le\_45  
feature with column\_index 6 = age\_ge\_55  
feature with column\_index 7 = age\_le\_55  
feature with column\_index 8 = age\_ge\_60  
feature with column\_index 9 = age\_le\_60  
feature with column\_index 10 = age\_ge\_65  
feature with column\_index 11 = age\_le\_65  
feature with column\_index 12 = age\_ge\_70  
feature with column\_index 14 = age\_ge\_75  
feature with column\_index 16 = age\_ge\_80  
feature with column\_index 38 = avg\_glucose\_level\_ge\_135  
feature with column\_index 66 = hypertension\_1

	classifier	accuracy	precision	recall	f1_score
1	Naive Bayes	0.732394	0.578947	0.702128	0.737827
4	Random Forest	0.739437	0.613636	0.574468	0.737167
5	CatBoost	0.739437	0.613636	0.574468	0.737167
2	Logistic Regression	0.732394	0.604651	0.553191	0.729206
0	kNN	0.725352	0.595238	0.531915	0.721158
6	XGBoost	0.718310	0.581395	0.531915	0.714954
3	Decision Tree	0.725352	0.611111	0.468085	0.714668





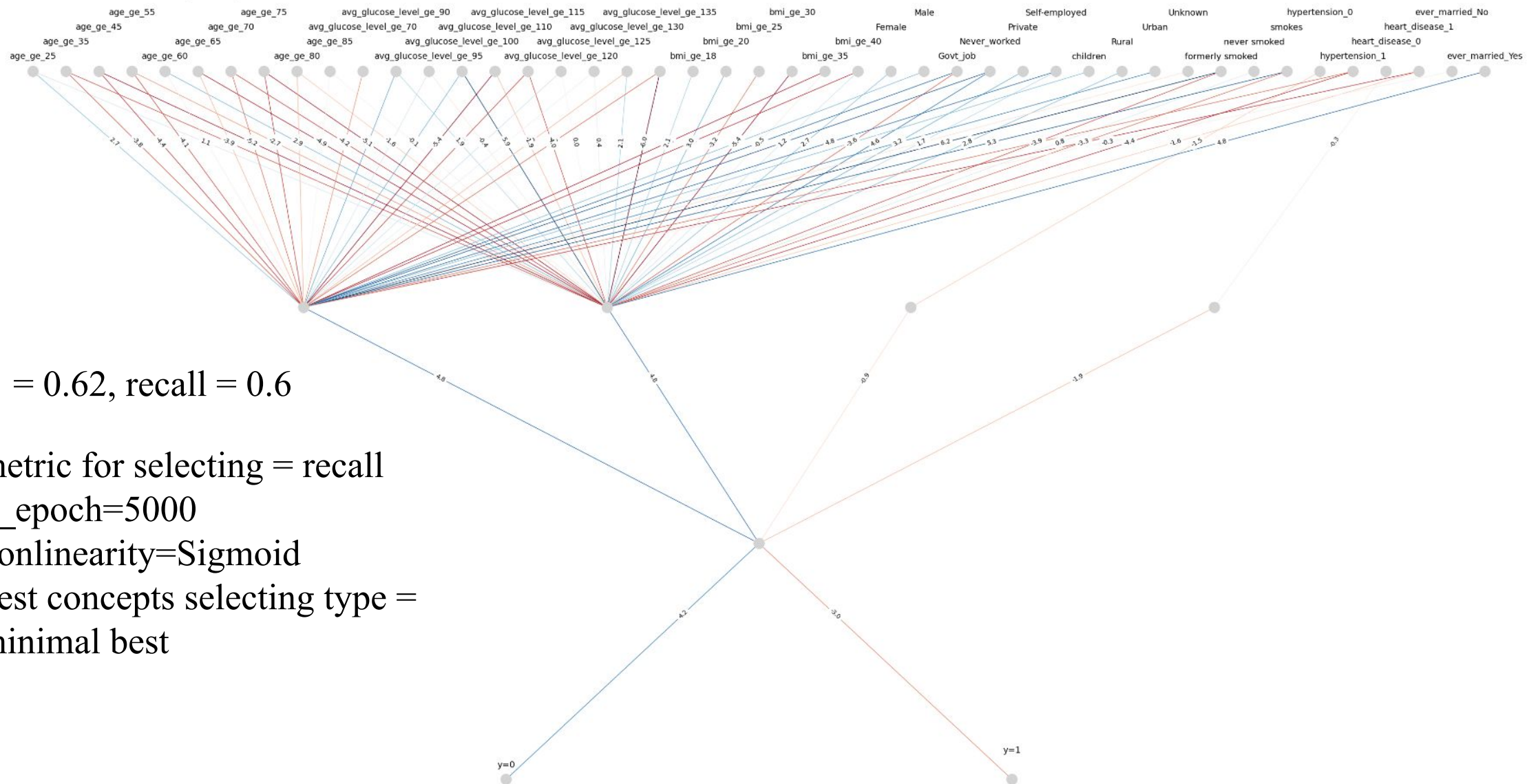
## CROSS VALIDATION

For each strategy, cross-validation was performed with the following parameters:

- metrics = [recall, f1\_score]
- n\_epoch = [2000, 5000, 7000]
- nonlinearity function = [Rely, Tanh, Sigmoid]
- best concepts selecting way = [best, best\_more, random]



Neural network with fitted edge weights



$f1 = 0.62$ ,  $\text{recall} = 0.6$

metric for selecting = recall

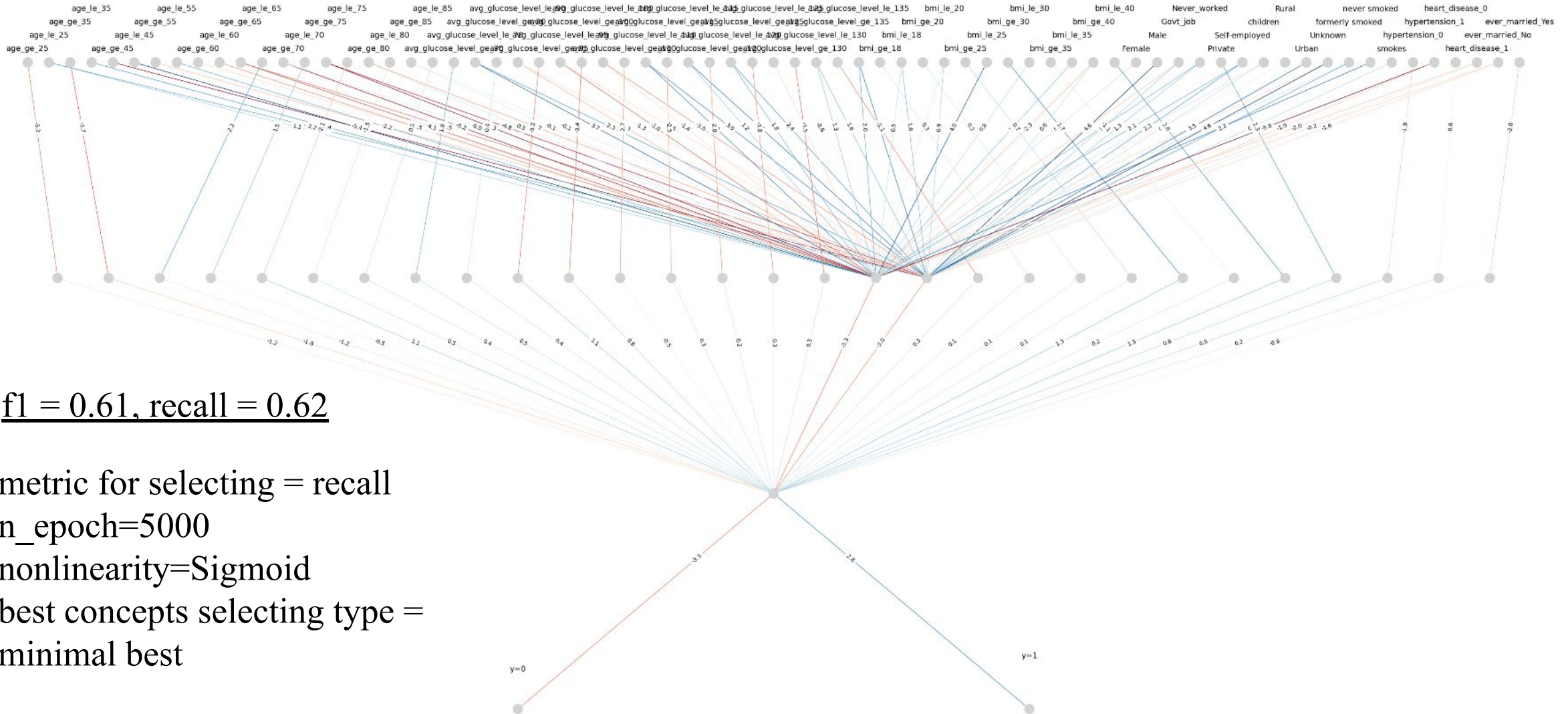
n\_epoch=5000

nonlinearity=Sigmoid

best concepts selecting type =  
minimal best



Neural network with fitted edge weights



$f1 = 0.61$ ,  $recall = 0.62$

metric for selecting = recall

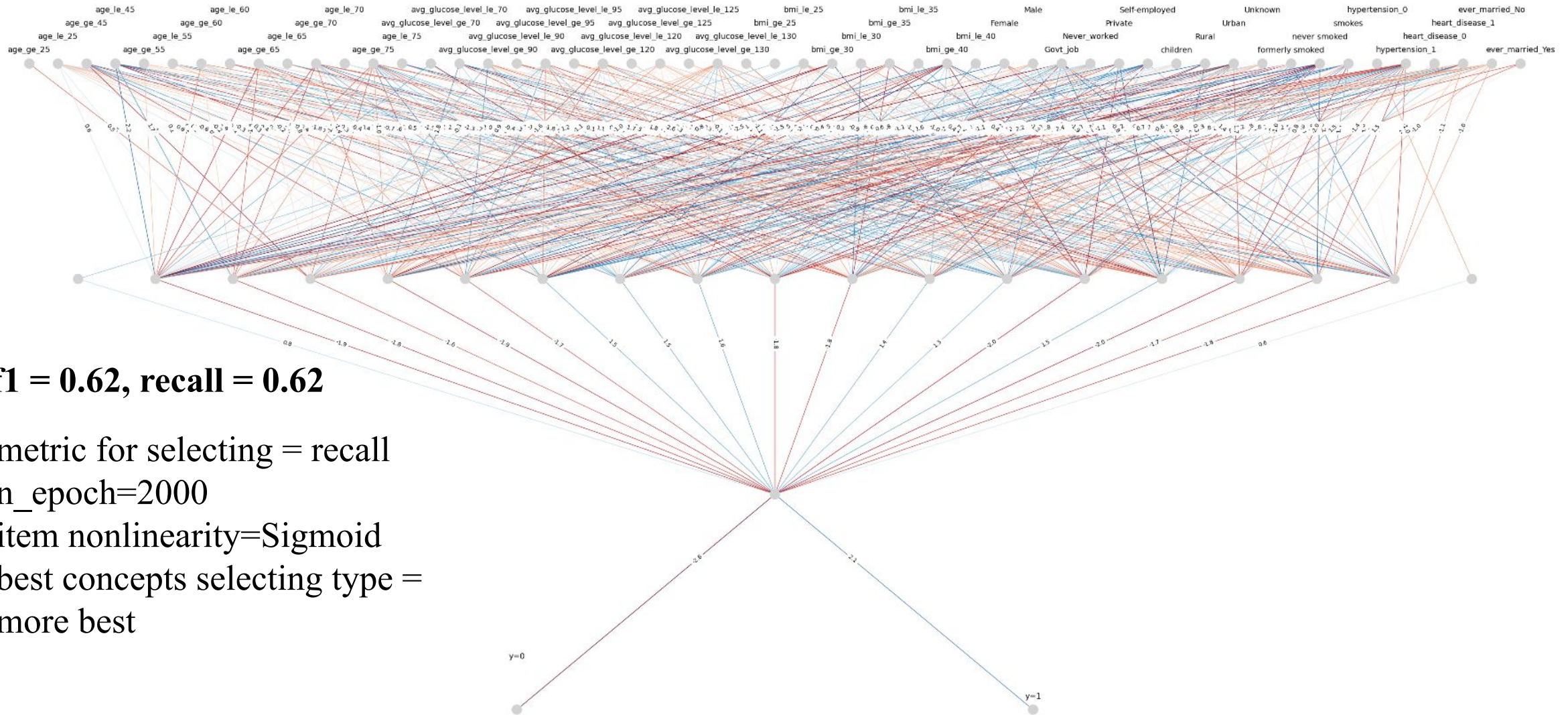
n\_epoch=5000

nonlinearity=Sigmoid

best concepts selecting type =

minimal best

Neural network with fitted edge weights



**f1 = 0.62, recall = 0.62**

metric for selecting = recall

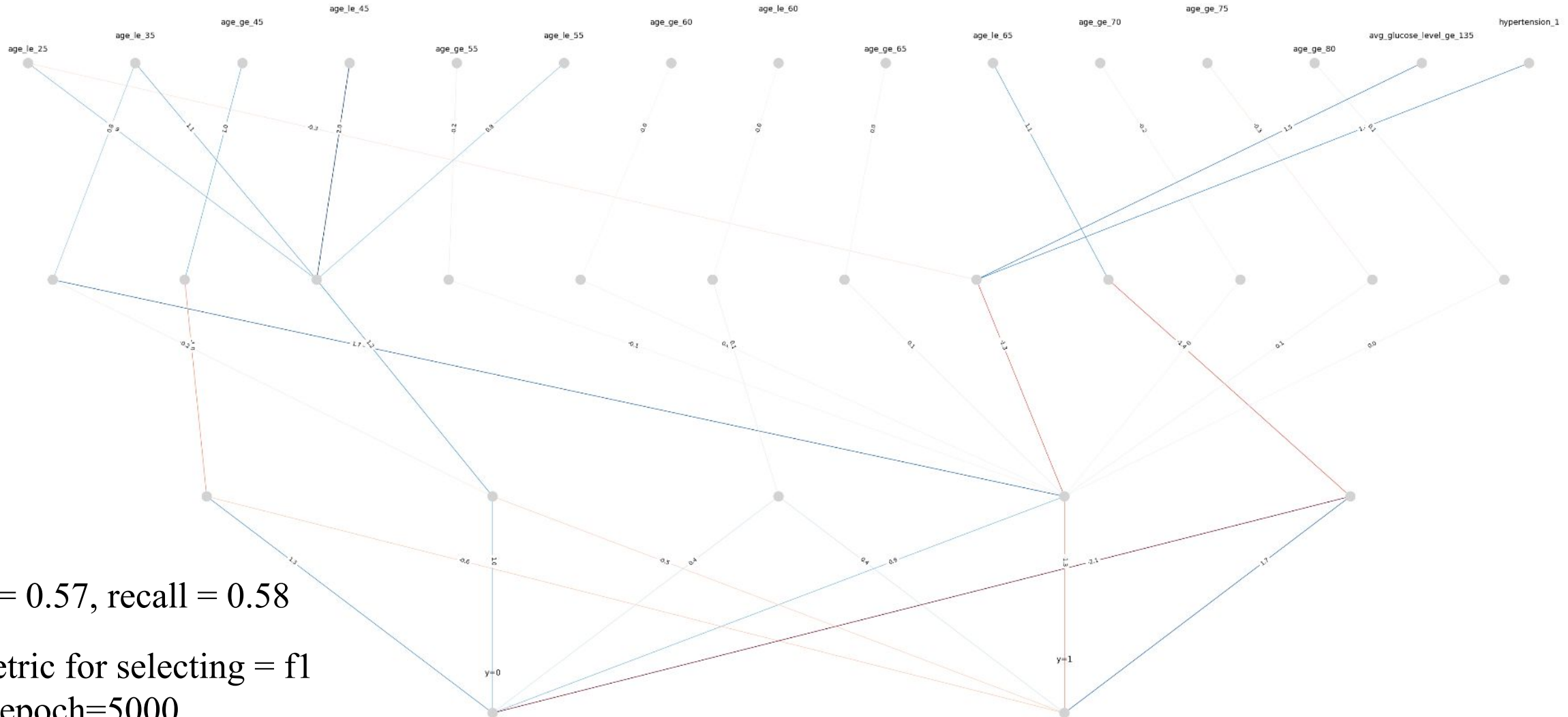
n\_epoch=2000

item nonlinearity=Sigmoid

best concepts selecting type =  
more best



Neural network with fitted edge weights



$f1 = 0.57$ ,  $\text{recall} = 0.58$

metric for selecting = f1

n\_epoch=5000

nonlinearity=ReLU, best concepts selecting type = random



## CONCLUSION

- The number of epochs depends on the dataset, the metric, and the method of adding nonlinearities (on average, the ideal value is 5000).
- The best metric is **recall** (it is suitable for most binarization strategies).
- All other hyperparameters (nnl function, method for concept selecting) need to be selected based on a validation sample (it is difficult to identify a common pattern for different experiments).

FCA models work at the level, and in some cases even better than standard machine learning algorithms, while significantly increasing the interpretability of the model.

THANK YOU FOR YOUR ATTENTION!