Ordered Sets for Data Analysis
Big HW: Neural FCA

Nasykhova Anastasia Artemovna

Link to the repo

Contents

1 Dataset overview **1**

2 Data preprocessing **2**
2.1 Binarization . . . 2

3 Baseline **3**
3.1 Metrics . . . 3
3.2 Performance . . . 4

4 FCA models **5**
4.1 Concept selecting method . . . 5
4.2 Nonlinearity function . . . 5
4.3 Number of epochs . . . 5
4.4 Cross validation hyperparameter tuning . . . 5
4.4.1 Binarization strategy №1 . . . 6
4.4.2 Binarization strategy №2 . . . 8
4.4.3 Binarization strategy №3 . . . 10
4.4.4 Binarization strategy №4 . . . 12
4.5 Best performance over all experiments . . . 13

5 Conclusion **13**

6 Appendix **13**

1 Dataset overview

Link to the dataset

Description: Stroke, a medical emergency that occurs due to the interruption of flow of blood to a part of brain because of bleeding or blood clots. According to the World Health Organization (WHO) stroke is the 2nd leading cause of death globally, responsible for approximately 11% of total deaths. Every year, more than 15 million people worldwide have a stroke, and in every 4 minutes, someone dies due to stroke. A stroke is generally a consequence of a poor style of living and hence, preventable in up to 80% of cases. Therefore, the prediction of stroke becomes necessary and should be used to prevent permanent damage by stroke.

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	9046	Male	67.0	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
1	51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	NaN	never smoked	1
2	31112	Male	80.0	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1
3	60182	Female	49.0	0	0	Yes	Private	Urban	171.23	34.4	smokes	1
4	1665	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.0	never smoked	1

Figure 1: Dataset information

Attribute information:

1. **id:** Unique identification number for each patient.
2. **gender:** "Male", "Female" or "Other"
3. **age:** The age of the patient.
4. **hypertension:** Whether the patient has hypertension (1 for yes, 0 for no).
5. **heart_disease:** Whether the patient has a history of heart disease (1 for yes, 0 for no).
6. **ever_married:** Whether the patient has been married (1 for yes, 0 for no).
7. **work_type:** The type of work the patient does (e.g., private, self-employed, government job, children, never work).
8. **Residence_type:** Whether the patient resides in an urban or rural area.
9. **avg_glucose_level:** The patient's average glucose level in blood.
10. **bmi:** Body Mass Index, measure of body fat based on height and weight.
11. **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).
12. **stroke:** The target variable indicating whether the patient had a stroke (1 for yes, 0 for no).

2 Data preprocessing

Missing values and uninformative columns: Due to the sufficient number of objects in the framework of the training project, there is no need to come up with a way to fill in rows with unfilled values, so they can simply be deleted. The id column does not carry semantic information, so it will also be deleted.

Class balancing: Due to the **huge** imbalance, it was decided to reduce the dataset to 700 lines (only with this ratio (1:2) on a reasonable number of epochs ($\tilde{3}000$), the FCA models stopped issuing constant predictions).

2.1 Binarization

1. **Dichotomic Scale**
   The columns hypertension, heart_disease, ever_married can have only two values (0 or 1), so the dichotomic scale will be used for them, that is, two columns will be created: has and does not has this attribute.
   Binarization will be performed using the get_dummies method from the pandas library.
2. **Nominal Scale**
   For the columns gender, smoking_status, work_type, Residence_type nominal scale will be used, since these are categorical signs and they can have more than 2 values.
   Binarization will be performed using the get_dummies method from the pandas library.
3. **Inter-Ordinal Scale**
   Features age, avg_glucose_level and bmi are numeric.
   Age significantly influences the incidence of strokes, with values ranging from 0 to 82 in the dataset. The condition is most commonly observed in individuals over the age of 45. In recent years, there has been a concerning trend towards younger patients, with an increasing number of younger adults experiencing strokes. Given this context, it is proposed to utilize broader age groupings at the beginning of the age spectrum, while implementing narrower groupings after the age of 45-50. Potential age groups could include: 0-25, 26-35, 36-45, 46-55, 56-60, 61-65, 66-70, 71-75, 76-80, 81-85, 86-90,

and 95+. Due to the significance of this variable, it is recommended to employ an inter-ordinal scale, which would involve establishing boundaries for the groups (25, 35, 45, 55, 60, 65, 70, 75, 80, 85, 90, 95) and applying the conditions of greater than or equal to ($\geq$) and less than or equal to ($\leq$).

The Body Mass Index (BMI) in the dataset ranges from 10 to 97. Various standards for categorizing BMI into groups are available in medical literature. These classifications typically include: underweight (2nd degree), underweight (1st degree), normal weight, overweight, obesity (1st degree), obesity (2nd degree), and obesity (3rd degree). This results in seven distinct groups: 0-18, 19-20, 21-25, 26-30, 31-35, 36-40, 40+; boundaries: 18, 20, 25, 30, 35, 40. Even when applying an inter-ordinal scale, this would generate 12 new features, which does not significantly increase computational complexity while preserving as much information as possible.

The glucose level in the dataset ranges from 55 to 271. Normal glucose levels vary based on age and sex, complicating the categorization process. The risk of stroke is significantly higher in patients with diabetes compared to those without the condition, with diabetes typically diagnosed at glucose levels of 115 or higher. In light of this, it is suggested to implement broader groupings in the tails and narrower groupings in the middle (boundaries: 70, 90, 95, 100, 110, 115, 120, 125, 130, 135, 140, 145, 150).

Binarization strategy №1: seemingly logical

- The older a person is, the more likely they are to have a stroke
- The higher the blood glucose level, the higher the probability of diabetes mellitus, the greater the probability of stroke
- The information differs from one source to another, but it is more common to find information that each increase in bmi by 5 points increases the risk of stroke by 21

Due to the fact that all signs have a direct relationship with the target variable, it seems logical and sufficient to use ordinal $\geq$.

Binarization strategy №2: inter-ordinal for each numeric

As part of the experiment, it would be interesting to also try to look at the inter-ordinal scaling method for all numeric features with the same breakdown boundaries.

Binarization strategy №3: larger intervals for numeric features

In contrast to the previous strategy, an inter-ordinal scale with fewer breakdowns will be used here for each numeric feature. Using this strategy, we will try to figure out whether it is worth saving as much information as possible or it is better to give the model the opportunity to generalize objects more strongly.

Binarization strategy №4: selecting features

Using the chi 2 test (SelectKBest(chi2, k=15))The 15 most valuable features will be identified after binarization according to the second strategy.

3 Baseline

In this research, eight standard classifiers (kNN, GaussianNB, Logistic Regression, Decision Tree, Random Forest, CatBoost, XGB) were used to provide a basic assessment of the effectiveness of four binarization strategies. Each model was configured using a selection of hyperparameters for cross-validation.

3.1 Metrics

Classifying a person as healthy in case they are actually at risk of stroke is a more serious problem than an error in classifying a healthy person as stroke patients. Although this is also undesirable, in practice it may be less dangerous to health.

However, using only this metric can lead to problems. For example, if you classify all samples as positive, the completeness will be equal to 1, but this does not mean that the classification is good in this case.

In this regard, F1-score was chosen as the main metric, as it takes into account both recall and precision. This allows you to get a more complete picture of the model's performance in predicting strokes.

3.2 Performance

	classifier	best_parameters	accuracy	precision	recall	f1_score
4	Random Forest	{'max_depth': 5, 'min_samples_leaf': 2, 'min_samples_split': 5, 'n_estimators': 100}	0.746479	0.641026	0.531915	0.739809
5	CatBoost	{'learning_rate': 0.01, 'max_depth': 3, 'n_estimators': 500}	0.739437	0.625000	0.531915	0.733583
6	XGBoost	{'learning_rate': 0.01, 'max_depth': 3, 'n_estimators': 500}	0.732394	0.609756	0.531915	0.727367
2	Logistic Regression	{'max_iter': 100, 'penalty': 'l1', 'tol': 0.01}	0.732394	0.615385	0.510638	0.725354
3	Decision Tree	{'max_depth': 3, 'min_samples_leaf': 1, 'min_samples_split': 2}	0.718310	0.581395	0.531915	0.714954
1	Naive Bayes	{}	0.704225	0.528090	1.000000	0.707928
0	kNN	{'n_neighbors': 7, 'weights': 'uniform'}	0.704225	0.553191	0.553191	0.704225

Figure 2: Binarization strategy №1

	classifier	best_parameters	accuracy	precision	recall	f1_score
5	CatBoost	{'learning_rate': 0.001, 'max_depth': 3, 'n_estimators': 100}	0.739437	0.613636	0.574468	0.737167
2	Logistic Regression	{'max_iter': 100, 'penalty': 'l2', 'tol': 0.01}	0.739437	0.625000	0.531915	0.733583
1	Naive Bayes	{}	0.718310	0.542169	0.957447	0.724388
0	kNN	{'n_neighbors': 7, 'weights': 'uniform'}	0.718310	0.574468	0.574468	0.718310
3	Decision Tree	{'max_depth': 3, 'min_samples_leaf': 4, 'min_samples_split': 2}	0.718310	0.581395	0.531915	0.714954
4	Random Forest	{'max_depth': 3, 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 50}	0.718310	0.589744	0.489362	0.710899
6	XGBoost	{'learning_rate': 0.01, 'max_depth': 3, 'n_estimators': 700}	0.718310	0.589744	0.489362	0.710899

Figure 3: Binarization strategy №2

	classifier	best_parameters	accuracy	precision	recall	f1_score
1	Naive Bayes	{}	0.767606	0.592105	0.957447	0.774072
4	Random Forest	{'max_depth': 7, 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 50}	0.746479	0.634146	0.553191	0.741717
5	CatBoost	{'learning_rate': 0.001, 'max_depth': 3, 'n_estimators': 500}	0.739437	0.613636	0.574468	0.737167
2	Logistic Regression	{'max_iter': 100, 'penalty': 'l1', 'tol': 0.0001}	0.739437	0.625000	0.531915	0.733583
6	XGBoost	{'learning_rate': 0.01, 'max_depth': 3, 'n_estimators': 500}	0.732394	0.609756	0.531915	0.727367
3	Decision Tree	{'max_depth': 3, 'min_samples_leaf': 4, 'min_samples_split': 2}	0.725352	0.611111	0.468085	0.714668
0	kNN	{'n_neighbors': 5, 'weights': 'uniform'}	0.704225	0.558140	0.510638	0.700702

Figure 4: Binarization strategy №3

	classifier	best_parameters	accuracy	precision	recall	f1_score
1	Naive Bayes	{}	0.732394	0.578947	0.702128	0.737827
4	Random Forest	{'max_depth': 3, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 50}	0.739437	0.613636	0.574468	0.737167
5	CatBoost	{'learning_rate': 0.001, 'max_depth': 5, 'n_estimators': 500}	0.739437	0.613636	0.574468	0.737167
2	Logistic Regression	{'max_iter': 100, 'penalty': 'l2', 'tol': 0.0001}	0.732394	0.604651	0.553191	0.729206
0	kNN	{'n_neighbors': 7, 'weights': 'uniform'}	0.725352	0.595238	0.531915	0.721158
6	XGBoost	{'learning_rate': 0.01, 'max_depth': 3, 'n_estimators': 500}	0.718310	0.581395	0.531915	0.714954
3	Decision Tree	{'max_depth': 3, 'min_samples_leaf': 4, 'min_samples_split': 2}	0.725352	0.611111	0.468085	0.714668

Figure 5: Binarization strategy №4

Best strategy number 3 with model Naive Bayes: f1_score = 0.77.

4 FCA models

4.1 Concept selecting method

To analyze the importance of choosing the best concepts, 3 methods were prepared:

- select_concepts_best is a standard selection algorithm (the minimum number of the best according to some metric required to describe all objects).
- select_concepts_more_best - in addition to the minimum number, threshold=10 next best concepts are selected from the standard algorithm.
- select_concepts_random - the number of concerts that was needed in the first algorithm is randomly selected (that is, if 10 concepts were needed to cover in the first algorithm, this algorithm will select 10 random ones).

4.2 Nonlinearity function

The basic methods were chosen for the experiment: ReLU, Tanh and Sigmoid.

4.3 Number of epochs

After trying to run the basic algorithm from the reference github, it became clear that the range of epochs should be 2000 - 7000. This is the ideal amount so that the model has time to learn to identify at least some dependencies, but has not yet had time to retrain into a constant classifier.

For self-written cross-validation, 3 options will be offered: 2000, 5000, 7000.

4.4 Cross validation hyperparameter tuning

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]

The output of all metrics for experiments can be viewed in the code, here I will give only the best model and parameter analysis for each binarization strategy.

4.4.1 Binarization strategy №1

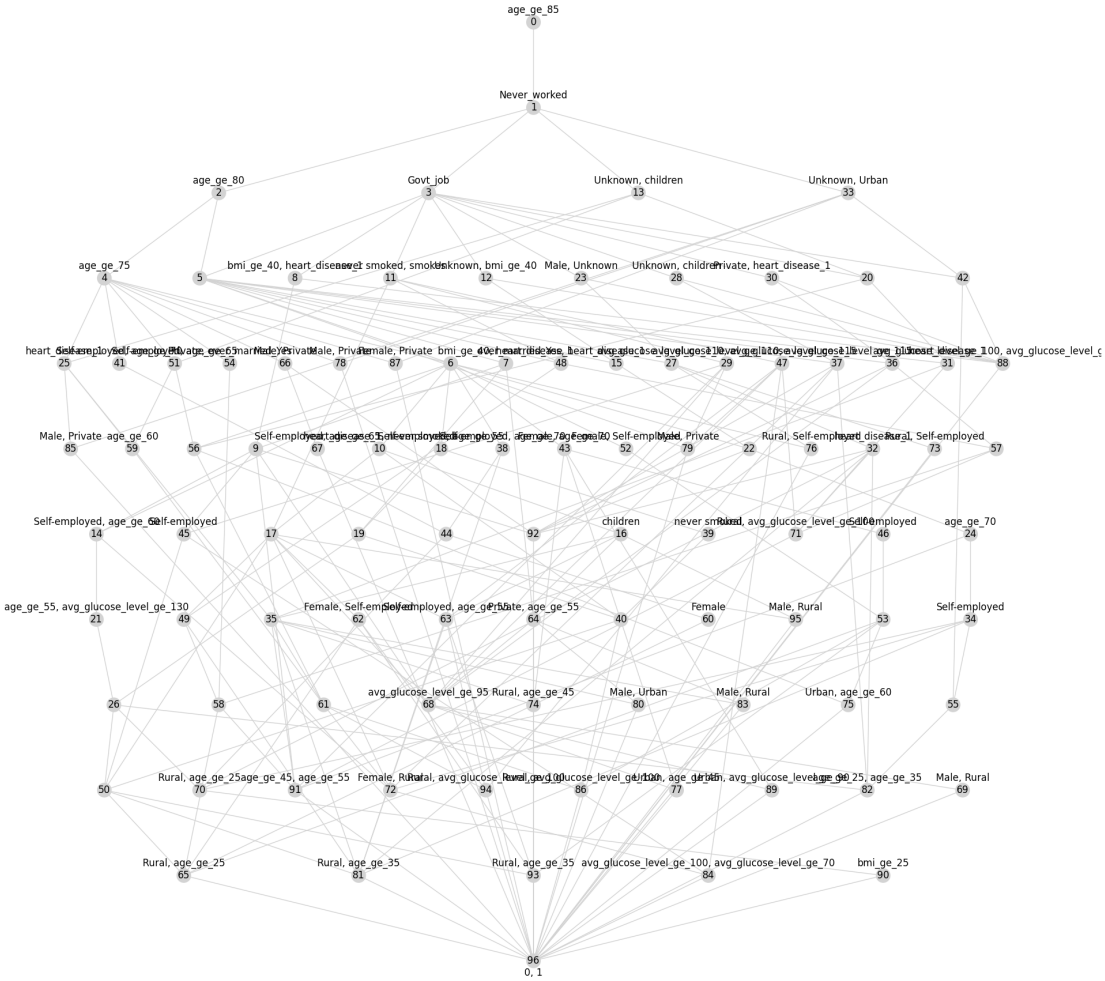

Figure 6: Concept Lattice №1

Best model parameters: metric for selecting = recall, n_epoch=5000, nonlinearity=Sigmoid, best concepts selecting type = minimal best.

Metric: f1 = 0.62, recall = 0.6

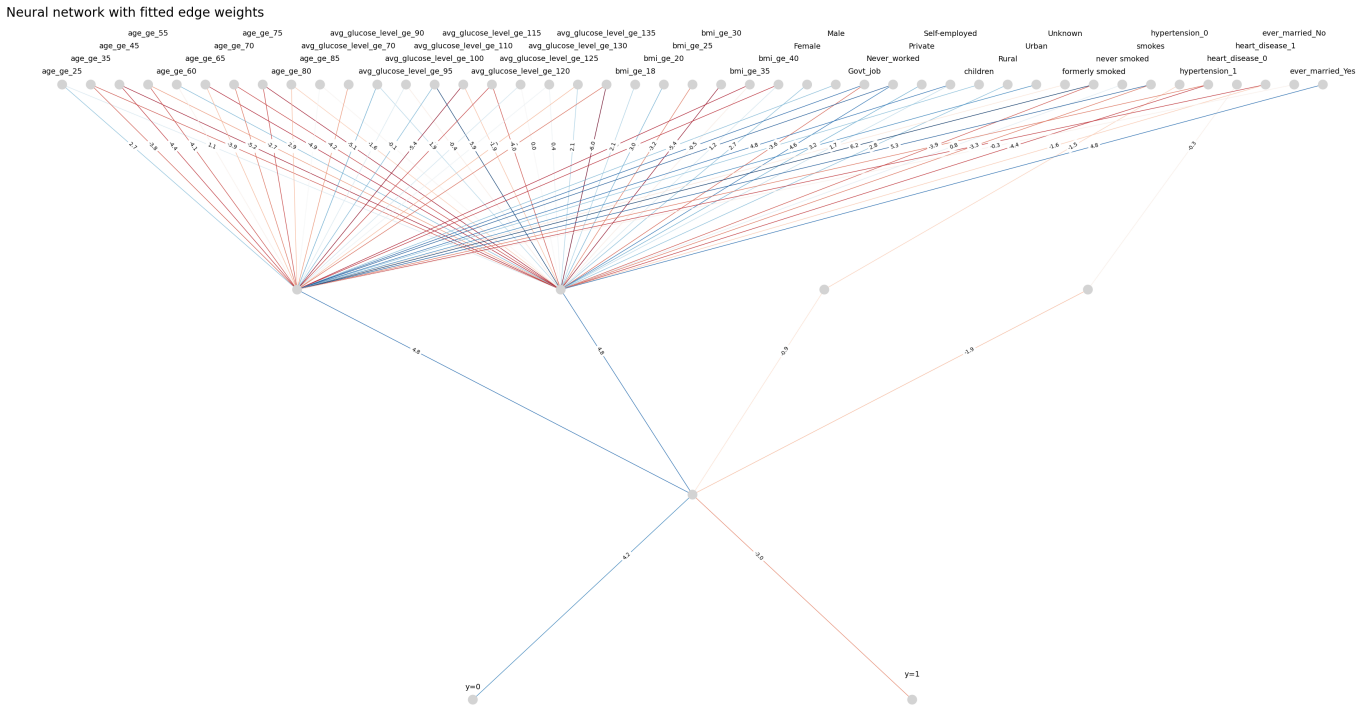

Figure 7: Concept Network №1

Interestingly, there are two neurons at the second level that generalize the largest number of features.

As expected, age¿=65, hypertension, smoking, and elevated blood glucose levels make the greatest contribution to increasing the likelihood of stroke.

A normal body mass index, normal glucose levels and, surprisingly, having a job (in particular a government job or your own business) reduces the likelihood of a stroke.

There is no obvious connection between stroke and gender, but according to different weights, it can be seen that men are more susceptible to them.

The situation is similar with the place of residence: people from the city have a high probability of stroke (which can be explained by environmental factors).

There is also no strong connection with the fact of marriage, but it is worth noting that those who were married have a positive weight at the edge (that is, closer to having a stroke). But this is also explained by logic - people who have been married, on average, are older than those who have never been married.

4.4.2 Binarization strategy №2

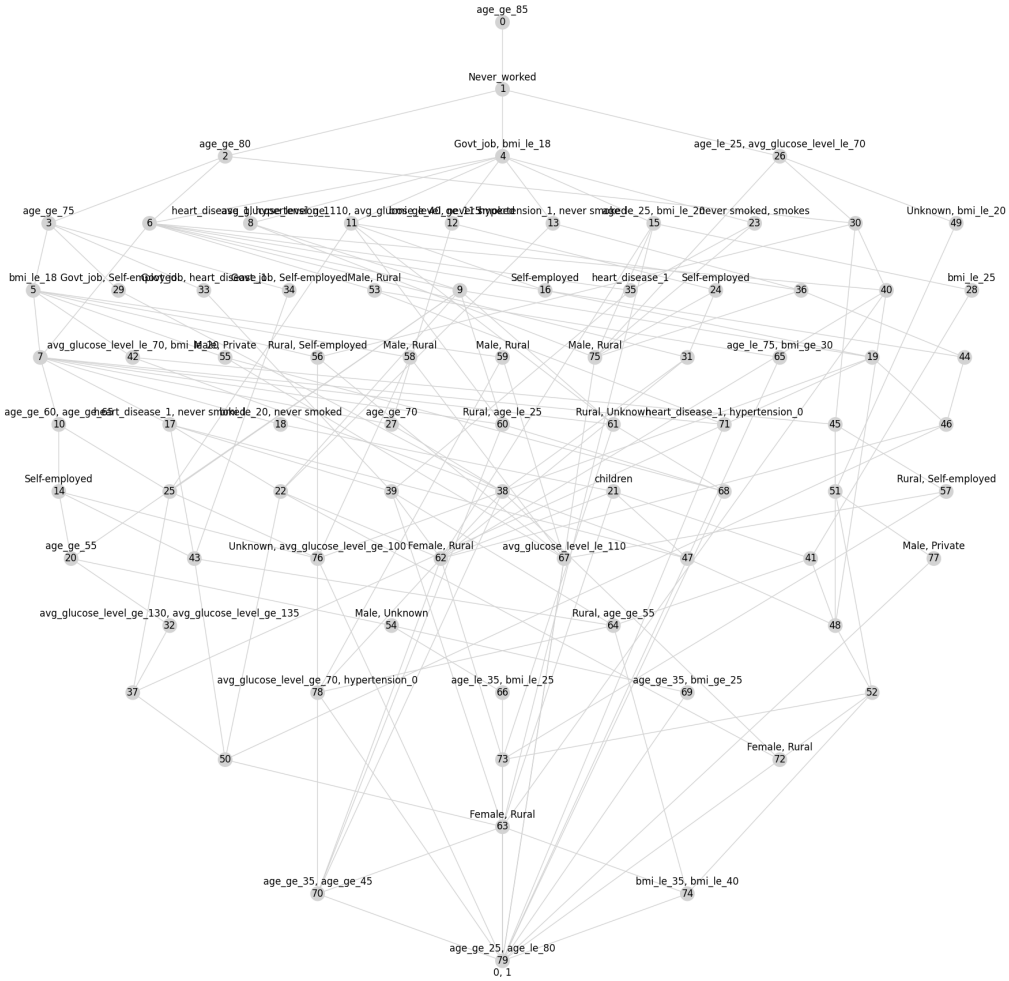

Figure 8: Concept Lattice №2

Best model parameters:

- metric for selecting = recall
- n_epoch=5000
- nonlinearity=Sigmoid
- best concepts selecting type = minimal best

Metric: f1 = 0.61, recall = 0.62

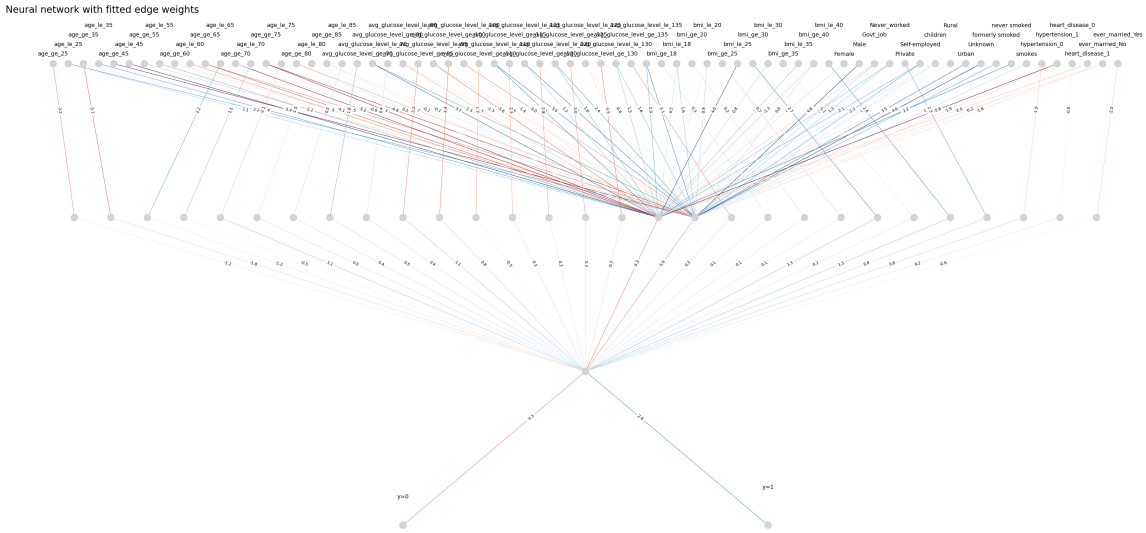

Figure 9: Concept Network №2

The influence of most of the signs remained about the same.

Of the interesting things, it can be noted that the weights of the signs are Male and Urban. Their weights have increased dramatically (now these are one of the main signs of a tendency to stroke)

Also, for this method of binarization, a sign that a person was once married reduces the likelihood of a stroke.

4.4.3 Binarization strategy №3

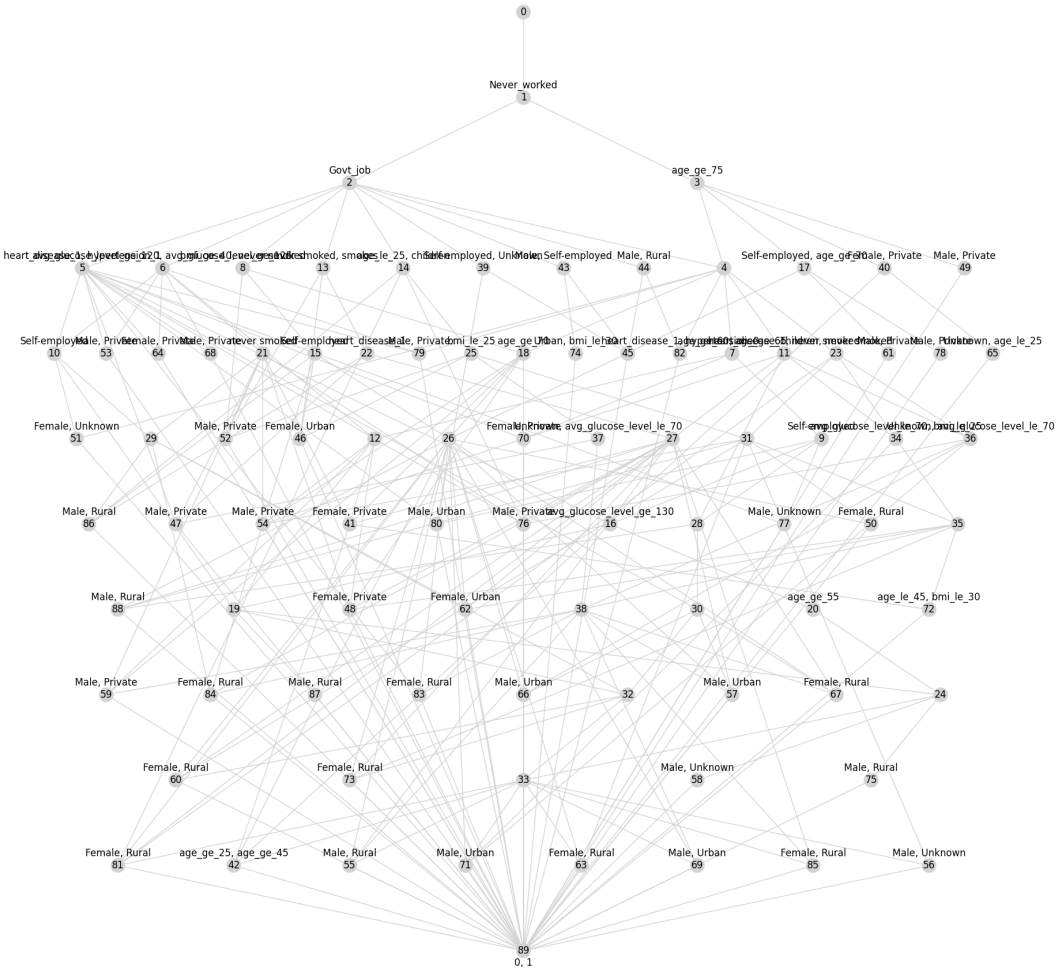

Figure 10: Concept Lattice №3

Best model parameters:

- metric for selecting = recall
- n_epoch=2000
- nonlinearity=Sigmoid
- best concepts selecting type = more best

Metric: f1 = 0.62, recall = 0.62

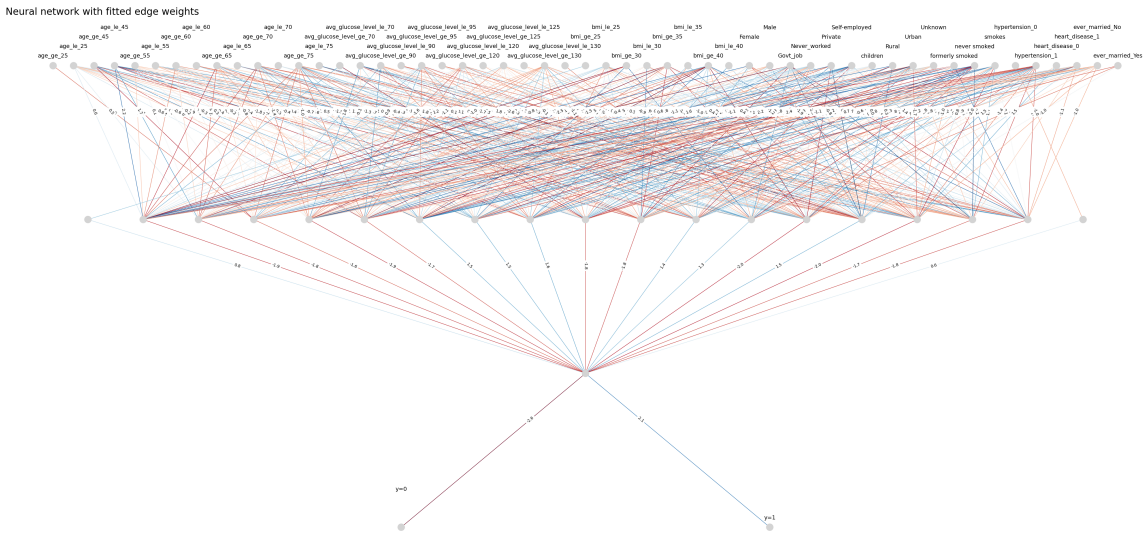

Figure 11: Concept Network №3

There are insanely many ribs... All interpretability is lost.

4.4.4 Binarization strategy №4

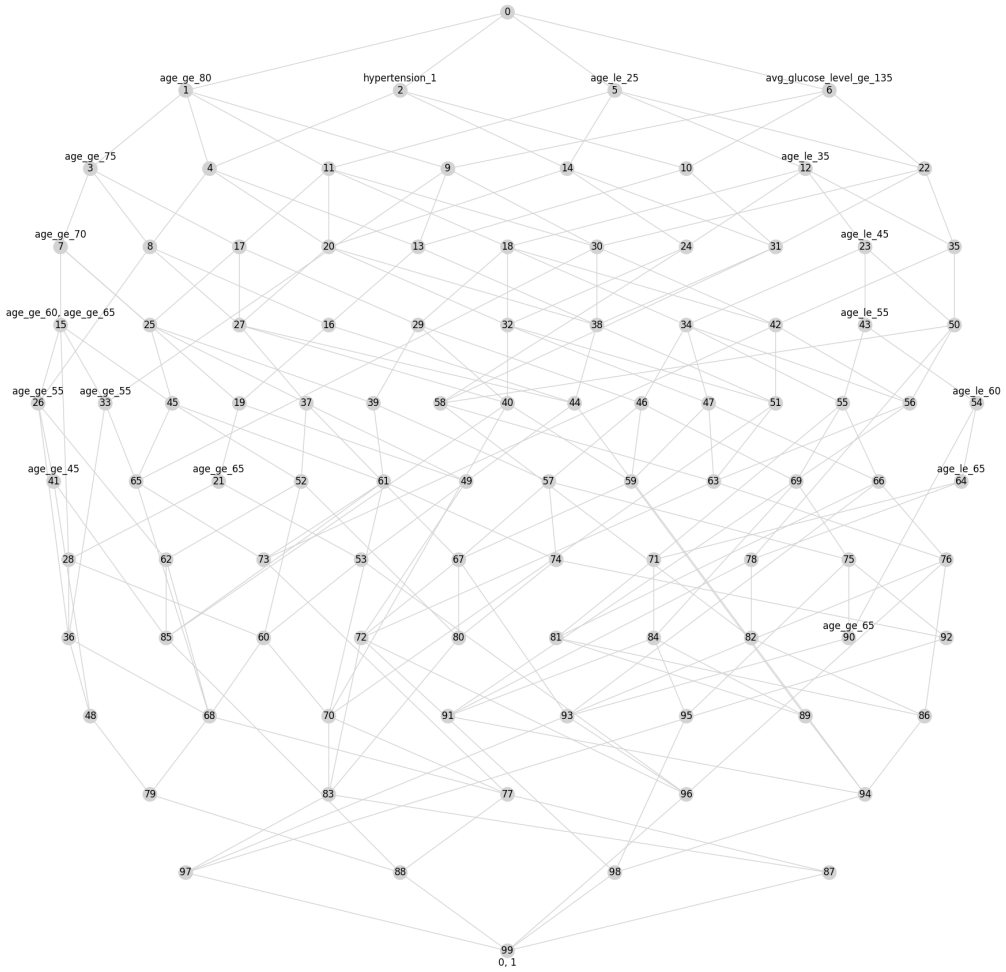

Figure 12: Concept Lattice №4

Best model parameters: metric for selecting = f1, n_epoch=5000, nonlinearity=ReLU, best concepts selecting type = random.
Metric: f1 = 0.57, recall = 0.58

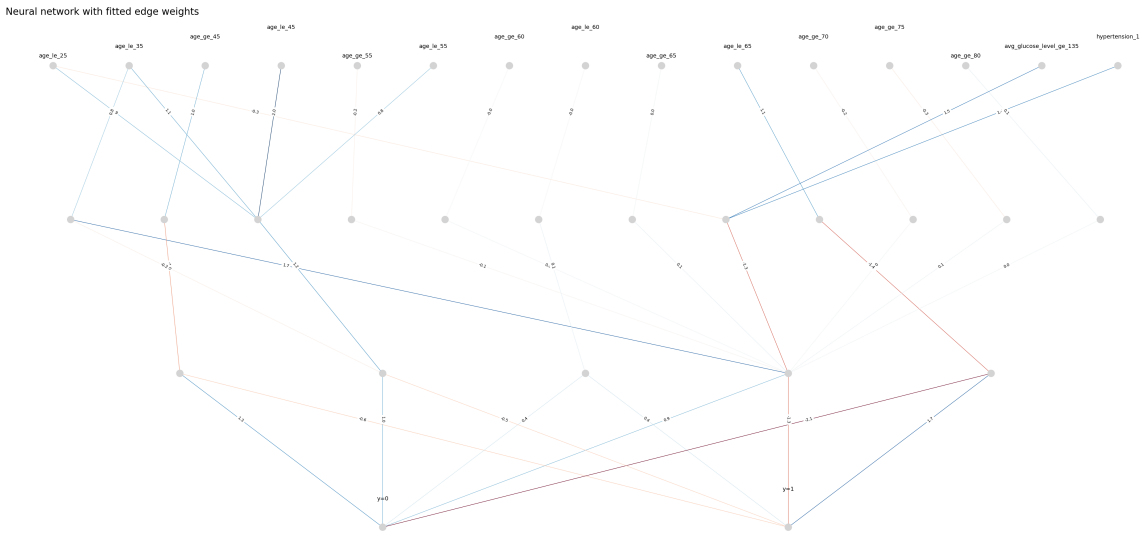

Figure 13: Concept Network №4

The most convenient for interpretation) Age $\leq$45 and, for some reason, increased glucose $\geq$ 135 also have the greatest influence in the direction of decreasing probability.

4.5 Best performance over all experiments

5 Conclusion

Problems encountered while working:

1. The CbO algorithm works for a very long time on a large sample volume.
2. Setting the right dependencies is a separate art.
3. An imbalance of classes (especially in the initial ratio of 1:23) can greatly spoil the situation (the model stops feeling dependencies and simply outputs a constant value).
4. The metric for choosing concepts is important (when choosing accuracy, the model became constant in any case).

Conclusions:

- 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!

6 Appendix

For convenience, the entire code is divided into subfiles with logic and each of them has a header system so that you can collapse some pieces of code

I'm sorry, if it would be convenient for everything to be in one file, I just didn't want to create an ipynb with 100 cells, it would be hard to navigate it.

Code structure:

- **docs** - folder with presentation and report
- **datasets** - folder with downloaded full dataset, resampled version with balanced classes and binarization for all strategies.
- **visualization** - folder for storing all the diagrams.
- **1. Preprocessing.ipynb** - code for preprocessing (classes balancing, binarization).
- **2. Base_models.ipynb** - code with the launch of 8 standard algorithms.
- **3.1 FCA 1st strategy.ipynb** - code with the search for ideal parameters for the Concept Network for the first binarization strategy.
- **3.2 FCA 2nd strategy.ipynb** - code with the search for ideal parameters for the Concept Network for the second binarization strategy.
- **3.3 FCA 3rd strategy.ipynb** - code with the search for ideal parameters for the Concept Network for the third binarization strategy.
- **3.4 FCA 4th strategy.ipynb** - code with the search for ideal parameters for the Concept Network for the fourth binarization strategy.