OSDA Big Homework:

Neural FCA

Aliev Mishan

National Research University "Higher School of Economics"

December 13, 2022

Dataset

Used dataset: openly available ([1]), with 6 attributes (5 binary and 1 numerical) and 2 target columns, with 120 rows.

The main goal of this dataset is to help predict two diseases of urinary system: an acute inflammations of urinary bladder (or cystitis in medical terms) and an acute nephritis.

Dataset

Dataset attributes and their values:

1. Temperature of patient - int values in the segment [35.5,41.5];
2. Occurrence of nausea - "yes" or "no";
3. Lumbar pain - "yes" or "no";
4. Urine pushing (continuous need for urination) - "yes" or "no";
5. Micturition pains - "yes" or "no";
6. Burning of urethra, itch, swelling of urethra outlet - "yes" or "no".

Our targets for binary classification:

1. Inflammation of urinary bladder (cystitis) - "yes" or "no";
2. Nephritis of renal pelvis origin - "yes" or "no".

In this work I will focus only on the prediction of cystitis.

Binarization strategy and prediction quality measure

Binarization strategy. According to [2], let's divide all data in column "Temperature of patient" into 2 groups (answers the question whether the patient has a temperature):

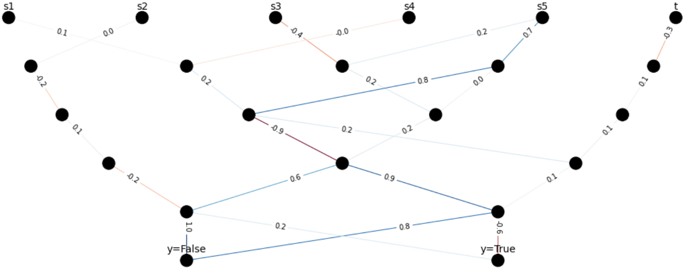
- "no" if temperature $\in$ [35.5,37.2];
- "yes" if temperature $\in$ [37.3,41.5].

Prediction quality measure. I prefer to use the **F1 score** because it maintains a balance between precision and recall for the classifier and it gives a better measure of the incorrectly classified cases than the accuracy metric.

First results

Important comment. Here and further I take the minimum possible count of best concepts and train neural networks 50000 epochs.

7 best concepts gives **F1 score** ≈ 0.67.



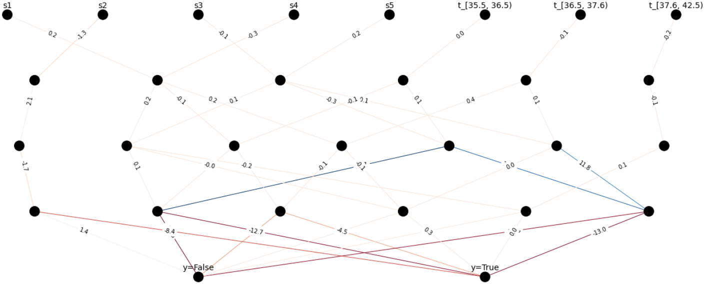
Neural network with fitted edge weights for model 1

Another attribute binarization

Based on [3], let's divide "Temperature of patient" into 3 groups:

- $t \in [35.5, 36.4]$;
- $t \in [36.5, 37.5]$.
- $t \in [37.6, 41.5]$.

12 best concepts gives **F1 score** ≈ 0.47.



Neural network with fitted edge weights for model 2

Another technique to select best concepts

Let's try to **select best concepts according to accuracy**. So, the results are:

- based on model 1 with **9** best concepts (let's name new model as model 3.1) - **F1 score** ≈ 0.51, **accuracy** ≈ 0.38;
- based on model 2 with **14** best concepts (let's name new model as model 3.2) - **F1 score** ≈ 0.59, **accuracy** ≈ 0.42.

Neural network graphs see in Appendix 1.

The efficiency of various nonlinearities to put in the network

I decide to use **Leaky ReLU** and **hyperbolic tangent** instead of base ReLU. So, the results are

1. based on model 1 for both nonlinearities we get **F1 score** $= 1$ (let's name them model 4 and model 5 for Leaky ReLU and hyperbolic tangent respectively - here we get **7** best concepts);
2. based on model 2 for both nonlinearities we get **F1 score** $= 1$ (similarly name them model 6 and model 7 - here we get **12** best concepts).

Neural network graphs see in Appendix 2.

Comparison of the prediction quality of the proposed model with State-of-the-Art approaches

I compare proposed model with results of **Logistic Regression** and **Random Forest Classifier** of default parameters.

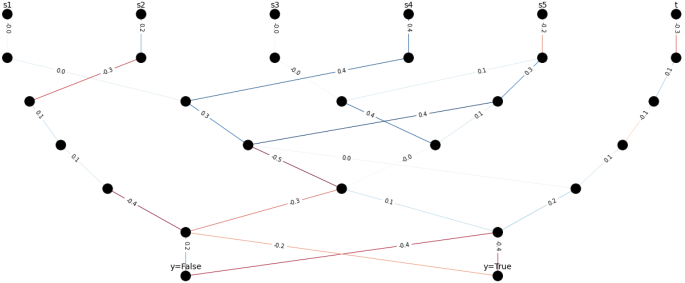
In both cases I get **F1 score** $= 1$ **BUT** these models work **faster** than proposed ones.

References:

1. https://archive.ics.uci.edu/ml/datasets/Acute+Inflammations
2. https://www.tuasaude.com/en/how-to-tell-if-you-have-a-fever/
3. https://en.wikipedia.org/wiki/Human_body_temperature
4. My code and dataset can be find in my GitHub repository https://github.com/thecrazymage/Neural-FCA

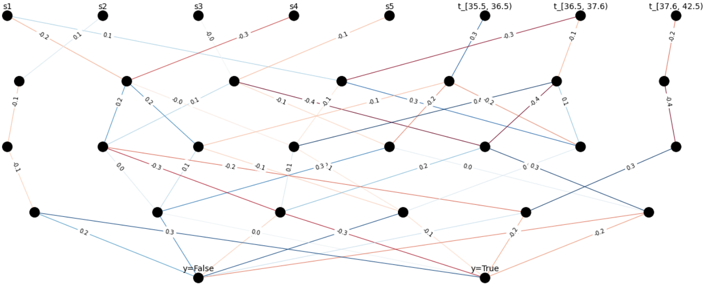
Thank You For Your Attention!

Appendix 1



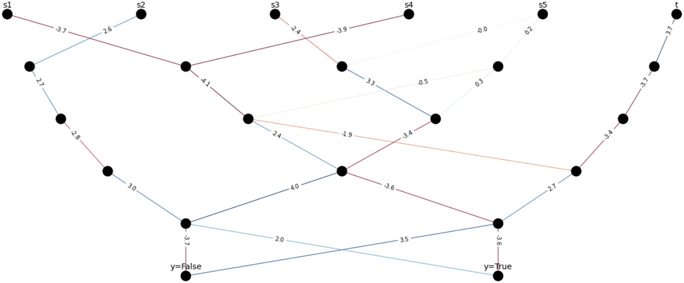
Neural network with fitted edge weights for model 3.1

Appendix 1



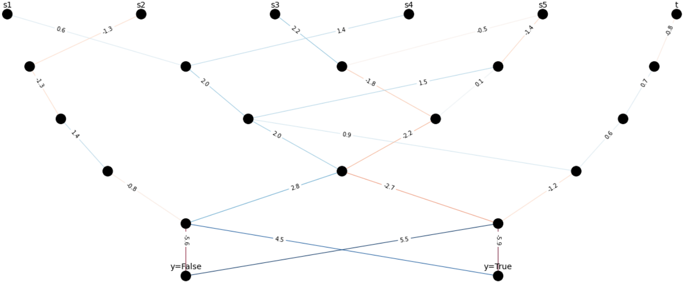
Neural network with fitted edge weights for model 3.2

Appendix 2



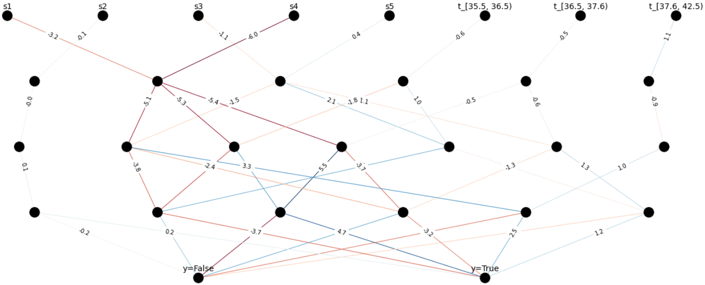
Neural network with fitted edge weights for model 4

Appendix 2



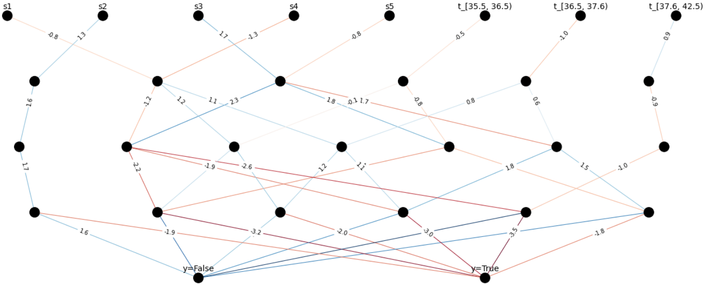
Neural network with fitted edge weights for model 5

Appendix 2



Neural network with fitted edge weights for model 6

Appendix 2



Neural network with fitted edge weights for model 7