OSDA Big Homework Report

Gorbunov Egor Ilyich

December 2024

1 Introduction

This is a report for the ”Neural FCA” big homework (project) for the course *Ordered Sets in Data Analysis*. A big homework's goal is to merge Neural Networks and Formal Concept Analysis (taught in Data Science Master program in HSE University, Moscow).

There is a need for interpretability in the AI field. It is usually hard to interpret the performance of neural networks. But there is an approach to create an interpretable neural network architecture based on the covering relation (graph of the diagram) of a lattice coming from monotone Galois connections. The vertices of such neural networks are related to sets of similar objects with similarity given by their common attributes, so easily interpretable. The edges between vertices are also easily interpretable in terms of concept generality or conditional probability.

The to-do list for this homework:

1. Choose a dataset (Kaggle, UCI, etc.), define the target attribute, binarize data and describe scaling (binarization) strategy for the dataset features.

2. Perform classification using standard ML tools.

3. Describe prediction quality measure best suited for the dataset (e.g. accuracy, F1 score, or any quality measure best suited for the dataset; fit and test the network on your task.

4. ry to improve the basic baseline with different ways.

5. Submit this report with comparison of all models, both standard and developed by you.

2 Dataset

I used a Credit Score Classification Dataset with 164 objects, 7 attribute columns (2 numerical, 5 categorical) and 1 target column. **The main goal** is to determine which credit score a client has (High/Average/Low). But I have decided to determine whether a client has a high score.

Columns (values):

- *Age* (25 - 53 years)
- *Gender* (Female/Male)
- *Income* (25000$ - 162500$)
- *Education* (”Bachelor’s Degree”, ”Master’s Degree”, ’Doctorate’, ’High School Diploma’, ”Associate’s Degree”)
- *Martial Status* (Single/Married)
- *Number of Children* (0, 1, 2, 3)
- *Home Ownership*(Rented/Owned)
- *Credit Score* (Low/Average/High) - OUR TARGET

Firstly, let’s look at numerical attributes’ distributions (*Age*/*Income*).

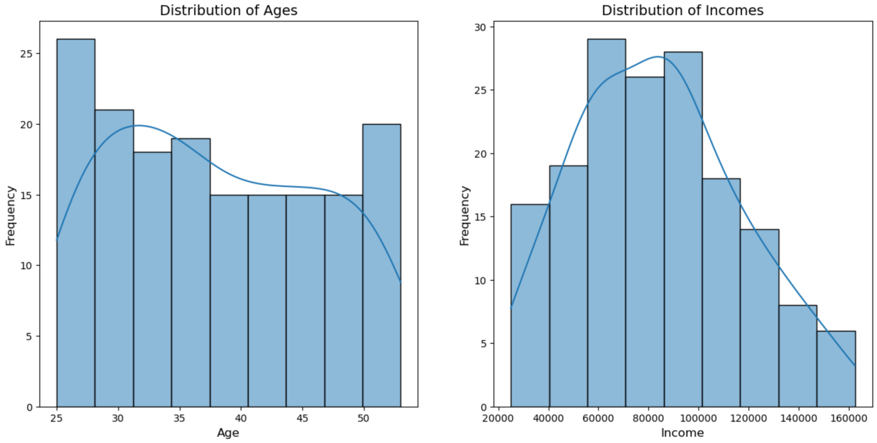

Figure 1: Distributions of Age/Income

Secondly, let’s look at categorical attributes.

Gender	Female	Male
Count	86	78

Table 1: *Gender* Distribution

Education	Bachelor	Master	Doctorate	High School	Associate
Count	42	36	31	30	25

Table 2: *Education* Distribution

Martial Status	Married	Single
Count	87	77

Table 3: *Martial Status* Distribution

№ of Children	0	1	2	3
Count	97	32	30	5

Table 4: *Number of Children* Distribution

Home Ownership	Owned	Rented
Count	111	53

Table 5: *Home Ownership* Distribution

Credit Score	High	Average	Low
Count	113	36	15

Table 6: *Credit Score* Distribution

Finally, due to the fact that the data has the class imbalance, let's make a new target attribute and drop the original one.

is Not High	False	True
Count	113	51

Table 7: Distribution of the target

It is a binary attribute where ”False” represents ”High” and ”True” represents ”Average or Low”.

3 Feature Engineering

Based on the distributions I decided to engineer the following features:

- *Age Group* (”25-33”/”34-43”/”44-53”)
- *Income Group* (”25k-74k”/”75k-124k”/”125k-174k”)
- *Has Children* (False - 0, True - 1/2/3)

Then, I dropped *Age*, *Income* and *Number of Children* because new attributes represent objects.

4 Binarization Strategy / Metric

Scales:

- *Age Group*, *Income Group*, *Education* - nominal scale
- *Gender*, *Martial Status*, *Home Ownership*, *Has Children* - dichotomic scale

After binarization there are 12 binary attributes, The train set is 75% of the whole data and the test set is 25%.

Metric. The dataset has 113 samples, which have a high credit score, and 51 samples, which have a low/average credit score. So, the most preferable metric for this date would be **F1 score**.

$$F1 = \frac{2 * precision * recall}{precision + recall}$$

5 First Results

16 best concepts give **F1 score** ≈ 0.962 on train set and **F1 score** ≈ 0.88 on test set. This CN was train on **2000 epochs**.

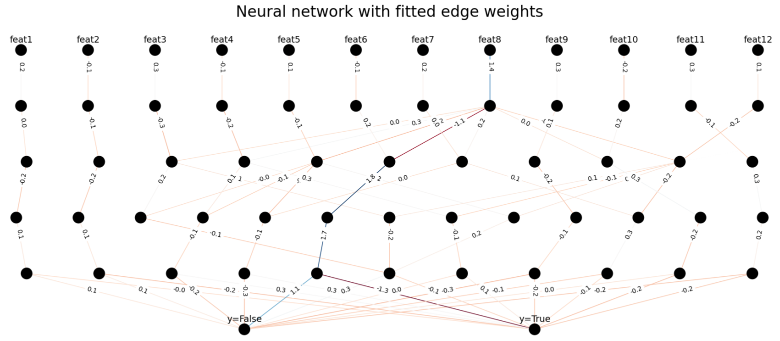

Figure 2: NN based on 16 best concepts from monotone concept lattice

6 Comparison with Standard Models

Now let's compare concept network's results with standard models (DT - Decision Tree, RF - Random Forest, LR - Logistic Regression, CB - CatBoost, XGB - XGBoost).

Model	DT	RF	KNN	LR	CB	XGB	**CN**
F1 Train	0.987	0.987	0.987	0.987	0.987	0.987	**0.926**
F1 Test	0.917	0.917	0.880	0.917	0.917	0.917	**0.880**

Table 8: Comparison

Now let's try to improve results of the concept network.

7 Modifications

Let's engineer *Age/Income* differently:

- *Age Group* ("25-40"/"40-53")
- *Income Group* ("25k-59k"/"60k-99k"/"100k-174k")

Then, let's use previous **scales:**

- *Income Group*, *Education* - nominal scale
- *Age Group*, *Gender*, *Martial Status*, *Home Ownership*, *Has Children* - dichotomic scale

After binarization there are 11 binary attributes.

9 best concepts give **F1 score** ≈ 0.962 on train set and **F1 score** ≈ 0.88 on test set. This CN was train on **2000 epochs**.

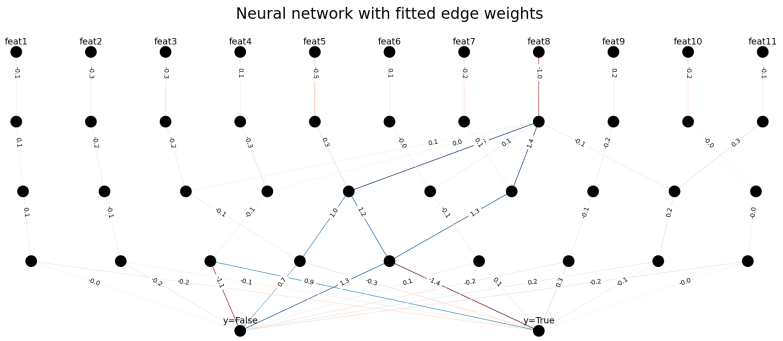

Figure 3: NN based on 9 best concepts from monotone concept lattice

F1 score is the same on train and test sets. But it takes less best concepts.

Let's try to change results with nonlinearities. *tanh* function from PyTorch gives the following results: **9 best concepts** give **F1 score** ≈ 0.987 on train set and **F1 score** ≈ 0.88 on test set. This CN was train on **2000 epochs**.

Although **F1 score** increased om the train set, it stayed the same on the test set. That could indicate an overfitting.

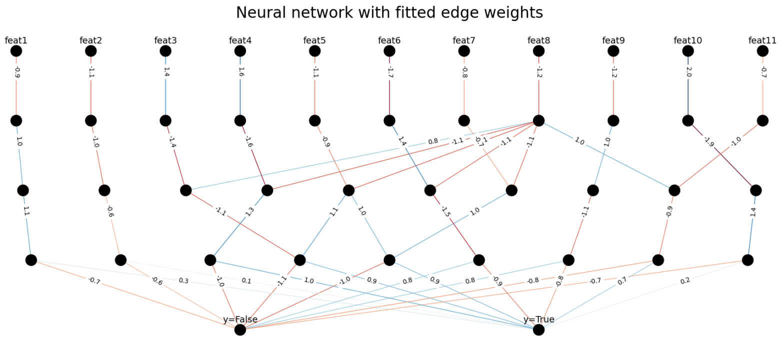

Figure 4: NN based on 9 best concepts from monotone concept lattice (with Tanh)

8 Conclusion

Although the merge between FCA and neural networks show a great performance on this dataset, other classical ML models outperform the CN. Even if interpretability is important to us, Decision Tree provides better scores and is easily interpretable.

9 References

1. Github repository of this project

2. Task formulation

3. Dataset on Kaggle