OSDA Big Homework Report, Neural FCA

Arsenev Vyacheslav

December 17, 2023

Contents

1 Introduction

For this research I choose 3 datasets: "Credit score classification", "Employee dataset" and "Lead Scoring Dataset". All of them was used to solve classical real, applied machine learning problems that usually occur in companies.

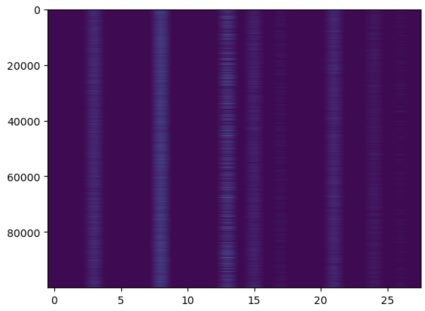
2 Credit score classification

For this dataset we try to predict rating of the customer in the bank.

2.1 Dataset

The dataset has longitude structure which means we have more then one observation for each customer (to be precise, we have 8 observations for each customer, sequential in time). Moreover we have 3 classes of target variable: "Good", "Standard", "Poor". Last but not least trail of data, that its raw and have lots of missing values(You can see it on Figure 1) and artefacts. But, lets deal with it step by step.

Figure 1: Missing values in Credit score classification dataset



Initial data contains 100000 rows and 28 columns, the columns descriptions has written below:

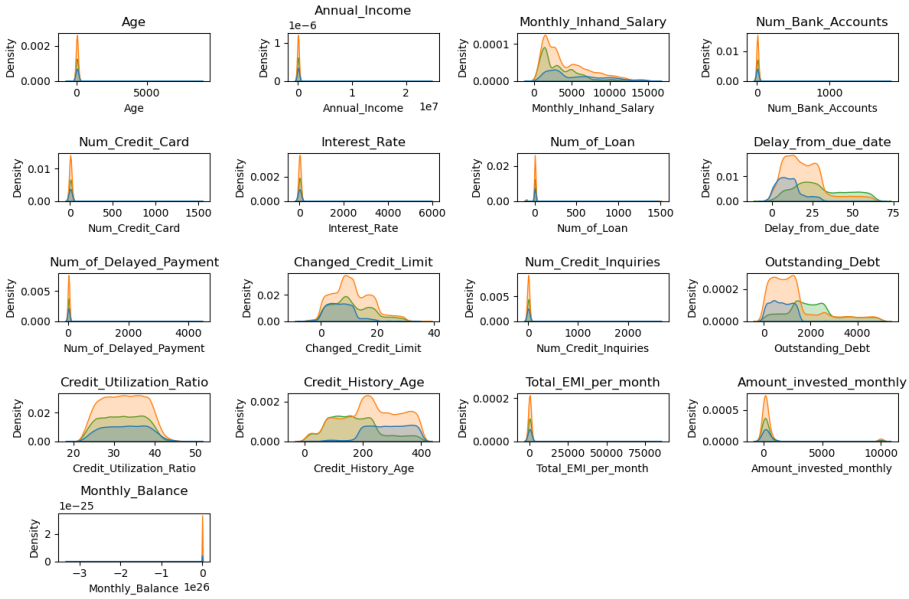
1. ID: Unique ID of the record - **dropped**
2. Customer_ID: Unique ID of the customer - **used for preprocessing but then dropped**
3. Month: Month of the year - **categorical\numerical**
4. Name: The name of the person - **dropped**
5. Age: The age of the person
6. SSN: Social Security Number of the person - **dropped**

7. Occupation: The occupation of the person - **categorical**
8. Annual_Income: The Annual Income of the person
9. Monthly_Inhand_Salary: Monthly in-hand salary of the person
10. Num_Bank_Accounts: The number of bank accounts of the person
11. Num_Credit_Card: Number of credit cards the person is having
12. Interest_Rate: The interest rate on the credit card of the person
13. Num_of_Loan: The number of loans taken by the person from the bank
14. Type_of_Loan: The types of loans taken by the person from the bank - **categorical**
15. Delay_from_due_date: The average number of days delayed by the person from the date of payment
16. Num_of_Delayed_Payment: Number of payments delayed by the person
17. Changed_Credit_Card: The percentage change in the credit card limit of the person
18. Num_Credit_Inquiries: The number of credit card inquiries by the person
19. Credit_Mix: Classification of Credit Mix of the customer - **categorical**
20. Outstanding_Debt: The outstanding balance of the person
21. Credit_Utilization_Ratio: The credit utilization ratio of the credit card of the customer
22. Credit_History_Age: The age of the credit history of the person
23. Payment_of_Min_Amount: Yes if the person paid the minimum amount to be paid only, otherwise no. - **categorical**
24. Total_EMI_per_month: The total EMI per month of the person
25. Amount_invested_monthly: The monthly amount invested by the person
26. Payment_Behaviour: The payment behaviour of the person - **categorical**
27. Monthly_Balance: The monthly balance left in the account of the person
28. Credit_Score: The credit score of the person - **target**

2.2 Prepossessing

The last column is target. "ID", "SSN" and "Name" are identifiers that not useful for predictions. "Customer_ID" is also identifier, but we hold it because we will use it when we input missing values. Since some of the columns are handwritten (its not mentioned in the description of dataset its my assumption) there is lots misprint such that: negative age, numbers containing "_" sign, outliers, etc.

Figure 2: Distribution of data

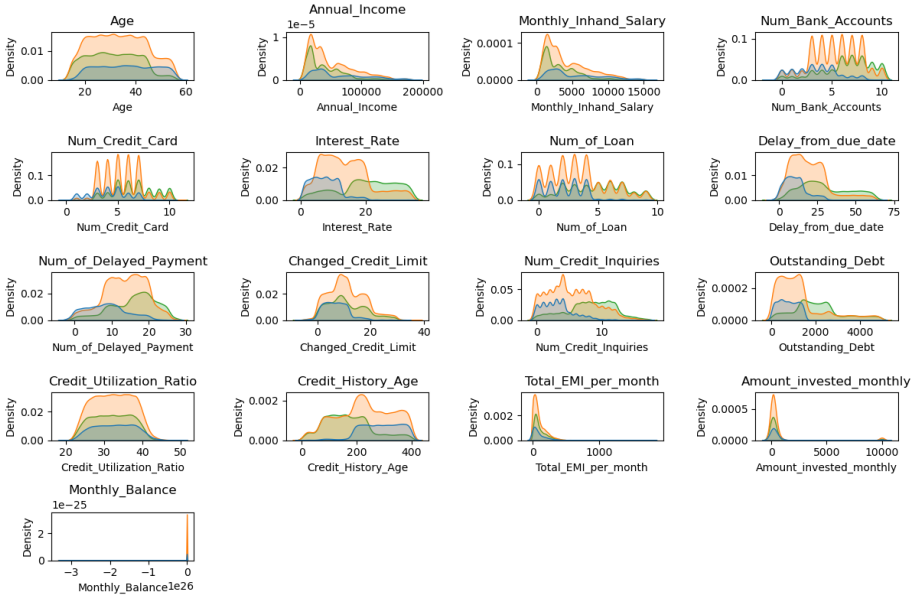


To detect outliers in single column, we compute the mode for each customer and find the maximum and minimum value among modes, then we remove (replace it to NaN) values that not belong to interval [min(mode), max(mode)]. This technique removes the most suspicious values. Why don't we remove outliers just by regular IQR method? Of course we can do it, but there is values that can be unusual to most of customers wheres it's regular value for certain customer. For example, most of the customers have 30k annual salary, but only few of them makes 100k per year(and it holds for all observations for this customer). If we use IQR without taking into account group structure we more likely obtain high amount of removed values.

Since we have longitude data, we use previous values as input for missing ones, its doesn't create endogenous problem. We just delete other observations where we couldn't apply this method (first value for customer in certain column is missing). After cleaning data we got 90602 observations, which means we lost

only 10% of examples.

Figure 3: Distribution of data after removing outliers



The next step is prepare string variable that contain type of loan (Type_of_Loan) we just encode it with One Hot Encoding (but it doesn't mean that for each person there is only one column equals to one). The last part of preparation data is binarization of target, I use model "One-vs-Others" and binarize data in this way: Poor as 1, Standard and Good as 0. Hence, our aim is to predict "bad cases". Indeed, detection of "bad cases" seems more important then distinguishing of two good cases.I also drop "Annual_Income" since it have correlation = 1 with Monthly_Inhand_Salary.

2.3 Machine Learning

In this part, I used Machine Learning models to predict classify whether customer at moment t has "Poor" credit score or not. For this purpose I use 4 models: Naive Bayes, Logistic Regression, Random Forest and Catboost. I use two settings: with lagged values(lagged target used too) and without lagged values.

2.3.1 Pool model

First set of models based on the assumption that all observations don't have impact on each other. I encode categorical features with one hot encoding and scale data. The result of training summed in Table 1.

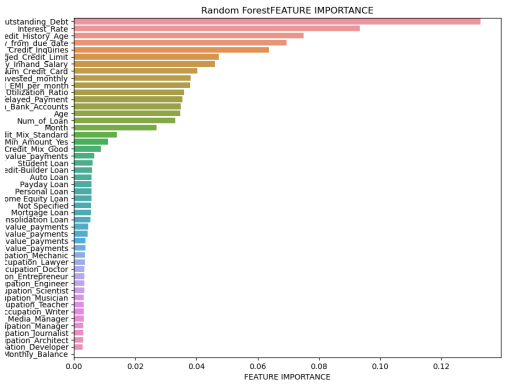
	Naive Bayes	Logistic regression	Random Forest	CatBoost
F1 score	0.539	0.652	0.792	0.738

Table 1: Pool model base classifiers results

We have imbalanced data: "Poor" = 26358, "Other" = 64244. So, for better performance I use simple oversampling method: I add new (64 244 - 26 358) = 37 886 "Poor" observations from the "Poor" distribution by selection with repetitions (bootstrap).

I change month from categorical one-hot-encoded feature to numerical one to reduce columns space and get better result. This changes in the pipeline gave me some extra points (for random forest i improved my result by 3% of F1 score). The features importance for Random Forest:

Figure 4: Feature Importance for Random Forest



2.3.2 Model with lags

In this part i tried to use first lag values for some of the features to improve models. I select features to lag according to mean coefficient of variety into the groups. So the list to lag is:

1. Credit_Score
2. Delay_from_due_date
3. Num_of_Delayed_Payment
4. Num_Credit_Inquiries
5. Changed_Credit_Limit
6. Credit_Utilization_Ratio

This method gave me better results in terms of quality. Models results summed in Table 2

	Naive Bayes	Logistic regression	Random Forest	CatBoost
F1 score	0.685	0.819	0.842	0.834

Table 2: Model with lags classifiers results

The Features Impertinence of Random Forest based on lagged Model:

2.3.3 Results

We obtained a good quality of the models, so lets sum up results before we move to Neural FCA .

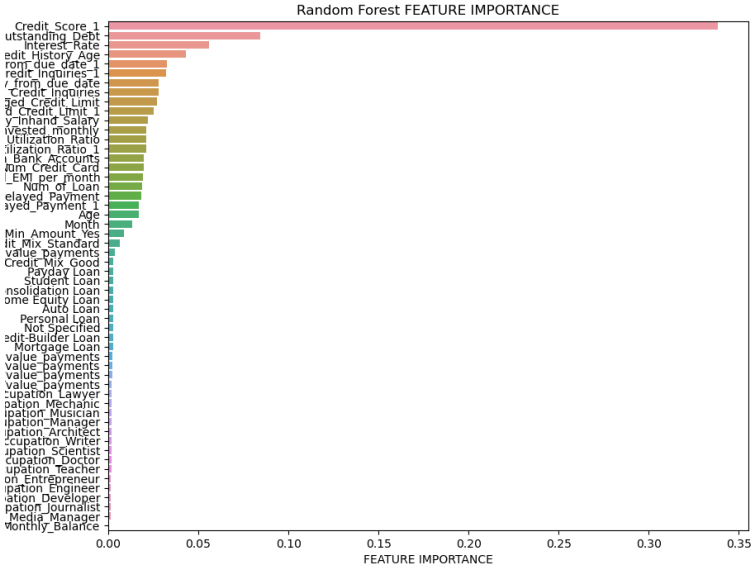
First, we tried don't use time structure in the data, that allow us make decisions for people after one month in the bank. On the other hand, if we use lags, only one lag of target column can predict about 75% of cases, but if we wont use lag of target, the other lagged variables don't make any improvements of the models. So, it up to us what model to choose and it depends on the circumstances.

I choose to use data without lags, because when we use credit score that was already gotten, we use expert knowledge more that we get from behavior of customers. Moreover, in the fiance sphere its more important to indicate bad caeses as soon as possible especially for new clients that have short history.

Feature importance from the Random Forest show us that only numerical features are important for prediction, so I choose only them to reduce sparsity of the data

Encoding month as an number improve quality of models. Oversampling also gave significant improvement.

Figure 5: Feature Importance for Random Forest with laged structure



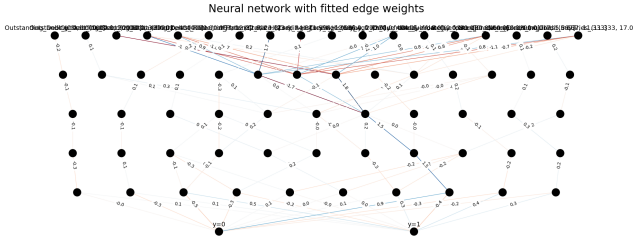
2.4 Neural FCA

In this part I build neural networks based on Formal Concepts for different set of features. I also try different binarization techniques and add non-linearity. To select concepts I use F1 score, and select first 15 to build the models.

2.4.1 Simple binarization

Our first strategy is just use 5 most important features. We divide every variable into 5 bins with the same width. With this strategy we get 25 columns. After training model we got this structure shown on **??**. We got F1 score = 0.647 on the test set.

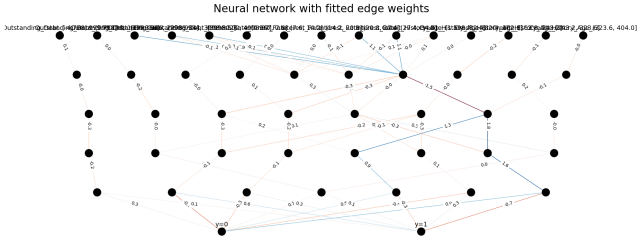
Figure 6: Fitted Network based on 5 features



2.4.2 Model based on 3 variables

For this model I tried to reduce number of features with the same binarization strategy. Fitted network is shown on Figure 7. For this model we obtained F1 score = 0.625.

Figure 7: Fitted Network based on 3 features



2.4.3 Quantile binarization

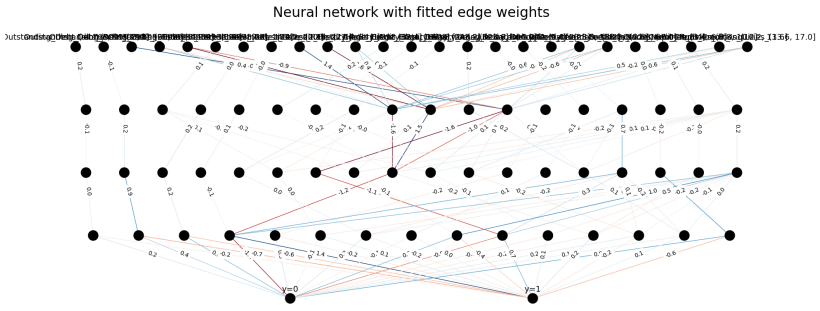
In this part I do more accurate binarization for two of variables and binarization based on quantiles for others features.

Binarization strategy:

1. For Delay_from_due_date we use such intervals: $\{(-\infty, 0), [0, 30), [30, 60), [60, 90), [90, \infty)\}$
2. For Outstanding_Debt we use such intervals: $\{(-\infty, 0), [0, 1000), [1000, 2000), [2000, 3000), [3000, 4000), [4000, \infty)\}$
3. For Interest_Rate', 'Credit_History_Age', 'Num_Credit_Inquiries' we use 3 bins quantile binarizaton: $\{[0, 0.33), [0.33, 0.66), [0.66, 1]\}$

Fitted model is shown below. For this model we got F1 score = 0.630.

Figure 8: Fitted network with based on quantile binarizaion



2.4.4 Adding Non-linearity

In this part i tried to add Tanh activation function for each model described above. This strategy.This strategy deteriorated the quality of all models.

2.4.5 Results

In this section I summed all neural models in Table 3.

Model	Num Features	Bin strategy	Activation	F1 score
1	5	Simple	ReLU	0.647
2	3	Simple	ReLU	0.625
3	5	Quantile	ReLU	0.630
4	5	Simple	Tanh	0.639
5	3	Simple	Tanh	0.626
6	5	Quantile	Tanh	0.614

Table 3: Neural FCA models results

Neural FCA shows simmilar result to classical models like Naive Bayes and Logistic Regression, but it is inferior to more complex models.

3 Lead Scoring Dataset

3.1 Dataset

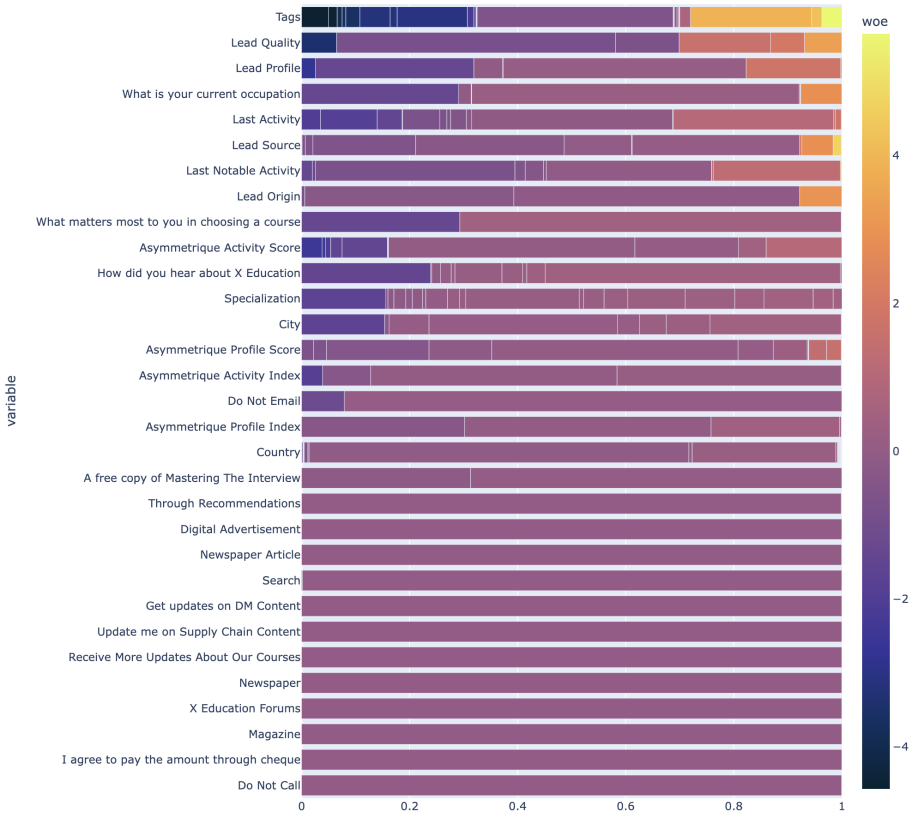
For this data set our purpose is to predict whether a user of study-platform buy paid curse or not.Initial data contains 9240 rows and 37 columns. The specific of this data is that it contains a lot of categorical variables and has small groups in this categorical variables and has.

3.2 Preprocessing

At the beginning I try to deal with categorical variables. To describe this variables i use Weight of Evidence (WOE) and Information Value (IV - a symmetric version of the Kullback-Leibler divergence). WOE can be used to illustrate importance of categorical variables and also can be good alternative for Target Encoder.

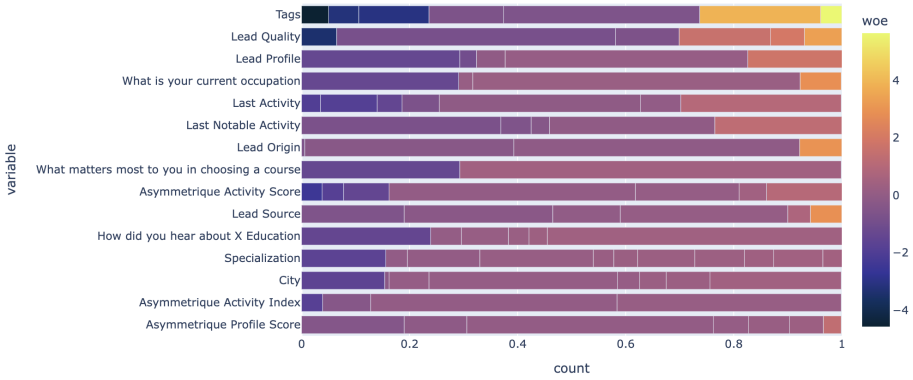
First we calculate WOE for each variable and stack and IV for each variable. Then we show the impotence of feature by stacking bar plot of sorted WOE for each variable (variables are sorted by IV)(Figure 9) and then we can "cut the bottom" of the plot to delete the most useless features.

Figure 9: Bar plot of categorical feature's WOE on initial dataset



Then after removing useless features, and merging small groups we obtain:

Figure 10: Bar plot of categorical feature's WOE on cleaned dataset



At the end we replace missing values for numerical features by KNN-inputer with k=2.

3.3 Machine Learning

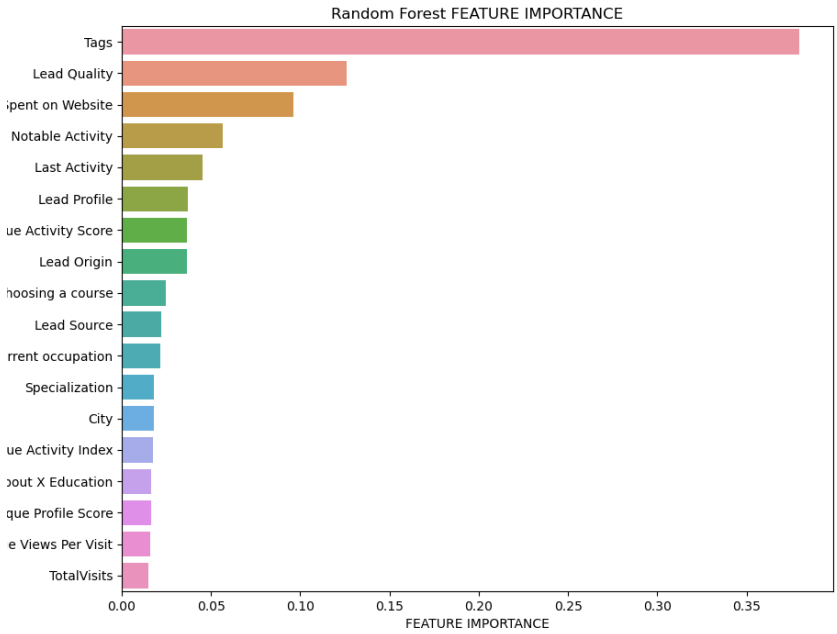
In this part a build several modals to classify users. I use the same set of models as before: Naive Bayes, Logistic Regression, Random Forest and Catboost. To input categorical features into models (except Catboost) i use WOE encoding, that i built by hand in the same way as Target Encoder implemented. We know that CatBoost has an advantage it can work with categorial features, so we use this opportunity and feed raw data to Catboost. Results of the trainig summed in Table 4.

	Naive Bayes	Logistic regression	Random Forest	CatBoost
F1 score	0.622	0.680	0.900	0.917

Table 4: classifiers results

We also select use feature importance of Random Forest(Figure 11) to reduce dimension in Neural FCA.

Figure 11: Feature Importance for Random Forest



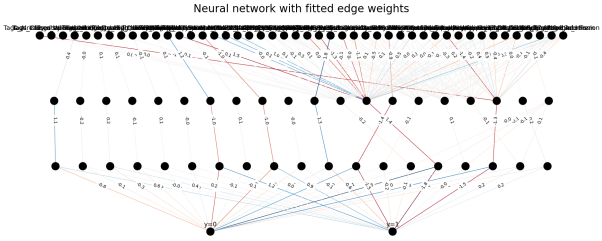
3.4 Neural FCA

In this part I use the same techniques as for first dataset. I use One Hot Encoder for categorical features and binarization for numerical. I also estimate Logistic Regression and Decision Tree on binarized data for better comparison.

3.4.1 Simple binarization

Our first strategy is just use all variables. We divide every variable into 5 bins with the same width. After training model we got this structure shown on **??**. We got F1 score = 0.798 on the test set.

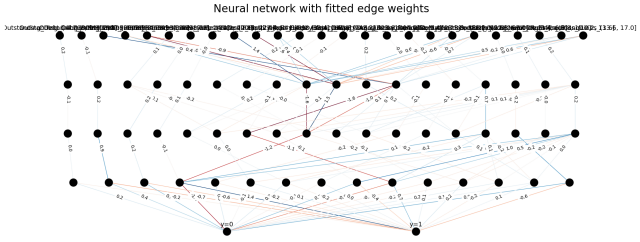
Figure 12: Fitted Network based on all features



3.4.2 Fitted Network based on 3 features

For this model I tried to reduce number of features with the same binarization strategy. Fitted network is shown on Figure 13. For this model we obtained F1 score = 0.768

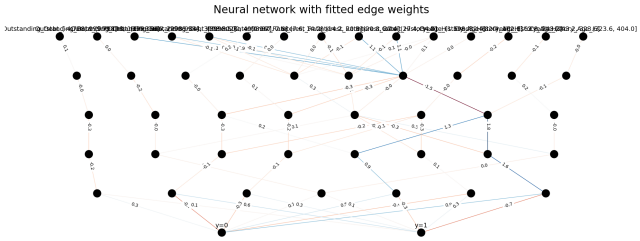
Figure 13: Fitted Network based on 3 features



3.4.3 Quantile binarization

In this part I hold number of bins equal to 5 but use quntile's bins insted of bins with the same width. Fitted network is shown on Figure 14. For this model we obtained F1 score = 0.772

Figure 14: Fitted Network based on 3 features



3.4.4 Adding Non-linearity

In this part i tried to add Tanh activation function for each model described above.This strategy significantly improved the quality of the models

3.4.5 Results

In this section I summed all neural models in Table 7.

Model	Num Features	Bin strategy	Activation	F1 score
1	18	Simple	ReLU	0.798
2	3	Simple	ReLU	0.768
3	18	Quantile	ReLU	0.772
4	18	Simple	Tanh	0.883
5	3	Simple	Tanh	0.828
6	18	Quantile	Tanh	0.885
Logistic Regression	18	Simple	-	0.892
Random Forest	18	Simple	-	0.908

Table 5: Neural FCA models results

As we can see for this dataset Neural FCA show comparable results even with SOTA models.

4 Employee dataset

4.1 Dataset

This dataset contains information about employees in a company, including their educational backgrounds, work history, demographics, and employment-related factors. It has 4653 rows and 9 columns. The data is very clean, so we don't need much preprossessing for it, so we move on ML part.

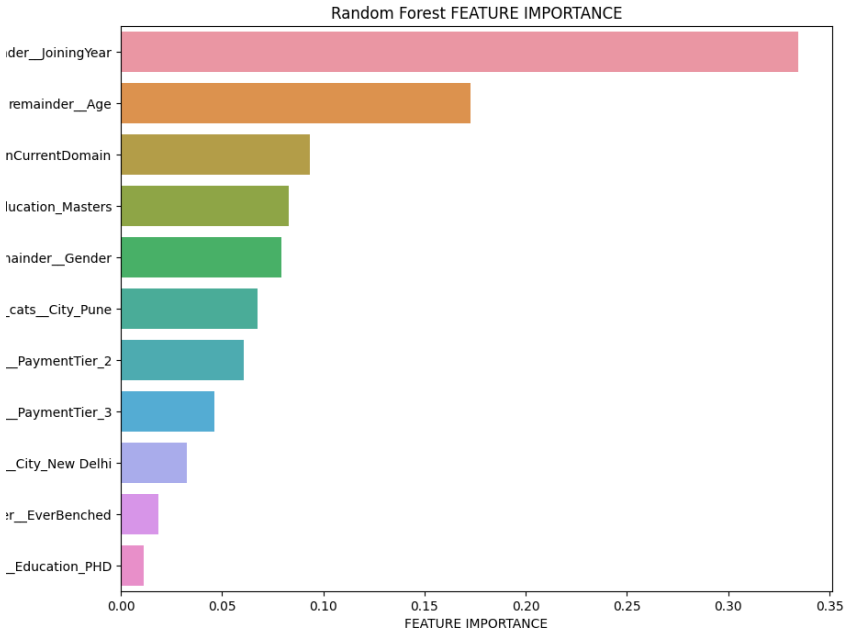
4.2 Machine Learning

We still use the same set of classifiers to predict target class: Naive Bayes, Logistic Regression, Random Forest and Catboost. We also use One hot Encoder for categorical features. The results of the training collected in Table 6, and importance of features for Random Forest are shown on **??**

	Naive Bayes	Logistic regression	Random Forest	CatBoost
F1 score	0.537	0.513	0.779	0.765

Table 6: Classifiers results

Figure 15: Feature Importance for Random Forest



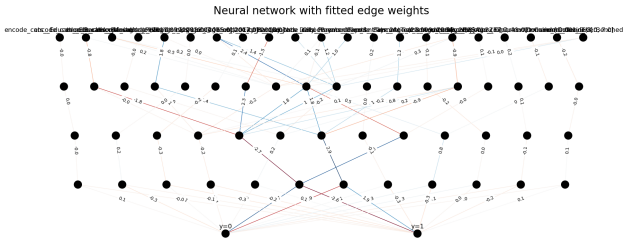
4.3 Neural FCA

In this part I use the same techniques as for first previous datasets.

4.3.1 Simple binarization

Our first strategy is just use all variables. We divide every numerical variable into 5 bins with the same width. After training model we got this structure shown on Figure 16. We got F1 score = 0.559 on the test set.

Figure 16: Fitted Network based on all features



This results isn't so good, and its looks quite expected: our network use only one edge (on last layer) to predict the target class.

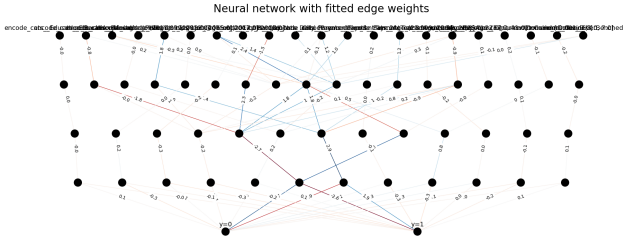
4.3.2 Quantile binarization

In this part I tried to use more accurate binning strategy to improve model results. I replace numerical variables with this quntile bins:

1. Experience In Current Domain = $\{[0, .5), [.5, 1.]$
2. Age = $\{[0, .25), [.25, .5), [.5, .75), [.75, 1.]$
3. Joining Year =$\{[0, .25), [.25, .5), [.5, .75), [.75, 1.]$

The fitted model is shown on Figure 17. For this model we obtained F1 score = 0.667

Figure 17: Fitted Network based on 3 features



4.3.3 Adding Non-linearity

In this part i tried to add Tanh and GELU activation functions to last model . This strategy significantly improved the quality of the models

4.3.4 Results

In this section I summed all neural models in Table 7.

Model	Num Features	Bin strategy	Activation	F1 score
1	19	Simple	ReLU	0.559
2	21	Quantile	ReLU	0.667
3	21	Quantile	Tanh	0.725
4	21	Quantile	GELU	0.665

Table 7: Neural FCA models results