Big Homework OSDA

Mukhanko Artem

19.12.2023

Table of contents

**Introduction**

In this project will be introduced implementation of a Lazy FCA classification algorithm based on pattern structures. Proposed algorithm was compared with popular ML models: Logistic Regression, Random Forest, KNeighbors, Decision Tree. You can find all code in my GitHub repository: https://github.com/artemm26/Lazy-FCA

**1) Selected Datasets**

I used the following datasets in the project:

Income dataset: Income Dataset (kaggle.com)

Stroke prediction dataset: Stroke Prediction Dataset (kaggle.com)

Employee future prediction: Employee Future Prediction (kaggle.com)

**1.1) Income dataset**

The dataset provided predictive feature like education, employment status, marital status to predict if the salary is greater than $50K.

Attribute Information:

1) age: (17 - 90 y.o.)
2) workclass: type of job
3) fnlwgt: final weight
4) education: (HS-grad, some-college…)
5) educational-num: education as Integer
6) marital-status: marital status
7) occupation
8) relationship: relationship status (husband, not-in-family…)
9) race (white, black)
10) gender (male, female)
11) capital-gain
12) capital-loss
13) hours-per-week
14) native-country

Dataset is shown below:

| | age | workclass | fnlwgt | education | educational-num | marital-status | occupation | relationship | race | gender | capital-gain | capital-loss | hours-per-week | native-country | income_>50K |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 67 | Private | 366425 | Doctorate | 16 | Divorced | Exec-managerial | Not-in-family | White | Male | 99999 | 0 | 60 | United-States | 1 |
| **1** | 17 | Private | 244602 | 12th | 8 | Never-married | Other-service | Own-child | White | Male | 0 | 0 | 15 | United-States | 0 |
| **2** | 31 | Private | 174201 | Bachelors | 13 | Married-civ-spouse | Exec-managerial | Husband | White | Male | 0 | 0 | 40 | United-States | 1 |
| **3** | 58 | State-gov | 110199 | 7th-8th | 4 | Married-civ-spouse | Transport-moving | Husband | White | Male | 0 | 0 | 40 | United-States | 0 |
| **4** | 25 | State-gov | 149248 | Some-college | 10 | Never-married | Other-service | Not-in-family | Black | Male | 0 | 0 | 40 | United-States | 0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **43952** | 52 | Private | 68982 | Bachelors | 13 | Married-civ-spouse | Exec-managerial | Husband | White | Male | 0 | 0 | 50 | United-States | 1 |
| **43953** | 19 | Private | 116562 | HS-grad | 9 | Never-married | Other-service | Own-child | White | Female | 0 | 0 | 40 | United-States | 0 |
| **43954** | 30 | Private | 197947 | Some-college | 10 | Divorced | Sales | Not-in-family | White | Male | 0 | 0 | 58 | United-States | 0 |
| **43955** | 46 | Private | 97883 | Bachelors | 13 | Never-married | Sales | Not-in-family | White | Female | 0 | 0 | 35 | United-States | 0 |
| **43956** | 30 | Private | 375827 | HS-grad | 9 | Never-married | Handlers-cleaners | Other-relative | White | Male | 0 | 0 | 40 | United-States | 0 |

After binarization:

| | age30_0 | age30_1 | age30_60_0 | age30_60_1 | age60_0 | age60_1 | educational-num10_0 | educational-num10_1 | educational-num20_0 | educational-num20_1 | ... | occupation_Tech-support | occupation_T |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | True | False | True | False | False | True | True | False | False | True | ... | False | |
| **1** | False | True | True | False | True | False | False | True | True | False | ... | False | |
| **2** | True | False | False | True | True | False | True | False | False | True | ... | False | |
| **3** | True | False | False | True | True | False | False | True | True | False | ... | False | |
| **4** | False | True | True | False | True | False | False | True | True | False | ... | False | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **43952** | True | False | False | True | True | False | True | False | False | True | ... | False | |
| **43953** | False | True | True | False | True | False | False | True | True | False | ... | False | |
| **43954** | False | True | True | False | True | False | False | True | True | False | ... | False | |
| **43955** | True | False | False | True | True | False | True | False | False | True | ... | False | |
| **43956** | False | True | True | False | True | False | False | True | True | False | ... | False | |

### 1.2) Stroke prediction Dataset

According to the World Health Organization (WHO) stroke is the 2nd leading cause of death globally, responsible for approximately 11% of total deaths. This dataset is used to predict whether a patient is likely to get stroke based on the input parameters like gender, age, various diseases, and smoking status. Each row in the data provides relavant information about the patient.

Attribute Information:

1) id: unique identifier
2) gender: "Male", "Female" or "Other"
3) age: age of the patient
4) hypertension: 0 if the patient doesn't have hypertension, 1 if the patient has hypertension
5) heart_disease: 0 if the patient doesn't have any heart diseases, 1 if the patient

has a heart disease
6) ever_married: "No" or "Yes"
7) work_type: "children", "Govt_jov", "Never_worked", "Private" or "Self-employed"
8) Residence_type: "Rural" or "Urban"
9) avg_glucose_level: average glucose level in blood
10) bmi: body mass index
11) smoking_status: "formerly smoked", "never smoked", "smokes" or "Unknown"*
12) stroke: 1 if the patient had a stroke or 0 if not

Dataset is shown below:

| | 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 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **5105** | 18234 | Female | 80.0 | 1 | 0 | Yes | Private | Urban | 83.75 | NaN | never smoked | 0 |
| **5106** | 44873 | Female | 81.0 | 0 | 0 | Yes | Self-employed | Urban | 125.20 | 40.0 | never smoked | 0 |
| **5107** | 19723 | Female | 35.0 | 0 | 0 | Yes | Self-employed | Rural | 82.99 | 30.6 | never smoked | 0 |
| **5108** | 37544 | Male | 51.0 | 0 | 0 | Yes | Private | Rural | 166.29 | 25.6 | formerly smoked | 0 |
| **5109** | 44679 | Female | 44.0 | 0 | 0 | Yes | Govt_job | Urban | 85.28 | 26.2 | Unknown | 0 |

After binarization:

| | age30_0 | age30_1 | age30_60_0 | age30_60_1 | age60_0 | age60_1 | avg_glucose_level55_0 | avg_glucose_level55_1 | avg_glucose_level70_160_0 | avg_glucose_le |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | True | False | True | False | False | True | True | False | True | |
| **2** | True | False | True | False | False | True | True | False | False | |
| **3** | True | False | False | True | True | False | True | False | True | |
| **4** | True | False | True | False | False | True | True | False | True | |
| **5** | True | False | True | False | False | True | True | False | True | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **5104** | False | True | True | False | True | False | True | False | False | |
| **5106** | True | False | True | False | False | True | True | False | False | |
| **5107** | True | False | False | True | True | False | True | False | False | |
| **5108** | True | False | False | True | True | False | True | False | True | |
| **5109** | True | False | False | True | True | False | True | False | False | |

### 1.3) Employee Future Prediction

A company's HR department wants to predict whether some customers would leave the company in next 2 years.

Attribute Information:

1) Education: education level

2) JoiningYear: year of joining company

3) City: city office, where posted

4) PaymentTier: (1- highest, 2-mid level, 3- lowest)

5) Age: current age

6) Genger (male, female)

7) EverBenched: ever kept out of projects for 1 month or more

8) ExperienceInCurrentDomain: experience in current field

9) LeaveOrNot: whether employee leaves the company in next 2 years

Dataset is shown below:

| | Education | JoiningYear | City | PaymentTier | Age | Gender | EverBenched | ExperienceInCurrentDomain | LeaveOrNot |
|---|---|---|---|---|---|---|---|---|---|
| 0 | Bachelors | 2017 | Bangalore | 3 | 34 | Male | No | 0 | 0 |
| 1 | Bachelors | 2013 | Pune | 1 | 28 | Female | No | 3 | 1 |
| 2 | Bachelors | 2014 | New Delhi | 3 | 38 | Female | No | 2 | 0 |
| 3 | Masters | 2016 | Bangalore | 3 | 27 | Male | No | 5 | 1 |
| 4 | Masters | 2017 | Pune | 3 | 24 | Male | Yes | 2 | 1 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 4648 | Bachelors | 2013 | Bangalore | 3 | 26 | Female | No | 4 | 0 |
| 4649 | Masters | 2013 | Pune | 2 | 37 | Male | No | 2 | 1 |
| 4650 | Masters | 2018 | New Delhi | 3 | 27 | Male | No | 5 | 1 |
| 4651 | Bachelors | 2012 | Bangalore | 3 | 30 | Male | Yes | 2 | 0 |
| 4652 | Bachelors | 2015 | Bangalore | 3 | 33 | Male | Yes | 4 | 0 |

After binarization:

| | Education_Bachelors | Education_Masters | Education_PHD | JoiningYear_0 | JoiningYear_1 | City_Bangalore | City_New Delhi | City_Pune | PaymentTier_1 | PaymentTie |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | True | False | False | False | True | True | False | False | False | Fa |
| 1 | True | False | False | True | False | False | False | True | True | Fa |
| 2 | True | False | False | True | False | False | True | False | False | Fa |
| 3 | False | True | False | False | True | True | False | False | False | Fa |
| 4 | False | True | False | False | True | False | False | True | False | Fa |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| 4648 | True | False | False | True | False | True | False | False | False | Fa |
| 4649 | False | True | False | True | False | False | False | True | False | T |
| 4650 | False | True | False | False | True | False | True | False | False | Fa |
| 4651 | True | False | False | True | False | True | False | False | False | Fa |
| 4652 | True | False | False | False | True | True | False | False | False | Fa |

## 2) Classification with standard ML tools

After selecting 3 datasets I proceed Machine learning models. As the preprocessing I fixed obvious errors in data, like deliting nan values and fixed datatypes in the datasets: formatted numerical to int and float datatypes; converted categorical features. For encoding categorical data, I used One-hot encoding. For numerical – interval scaling.  This preprocessing step aimed to enhance the overall quality of the datasets, ensuring they were well-structured and ready for consumption by a variety of classification algorithms.

The models were used: Logistic Regression, Random Forest, KNeighbors, Decision Tree. They have been selected for their widespread usage and proven performance in classification tasks. In this project, I employed accuracy score and F1-score as key metrics for evaluating the performance of classification models across three datasets. Accuracy score represents the ratio of correctly predicted instances to the total instances, providing a general overview of a model's correctness. On the other hand, F1-score balances precision and recall, making it particularly valuable when dealing with imbalanced datasets. It considers both false positives and false negatives, offering a more comprehensive assessment of a model's effectiveness in scenarios where different misclassification types bear varying consequences.

The following tables present how the machine learning models performed on the three different datasets. It includes accuracy results, recall 0, recall 1 and F1 scores for each model.

Income prediction:

| | Model | Accuracy | Recall_0 | Recall_1 | F_score |
|---|---|---|---|---|---|
| 0 | LogisticRegression | 0.8167 | 0.9035 | 0.5417 | 0.5865 |
| 1 | RandomForest | 0.8333 | 0.8816 | 0.6806 | 0.6622 |
| 2 | KNeighbors | 0.8167 | 0.8553 | 0.6944 | 0.6452 |
| 3 | DecisionTree | 0.7900 | 0.8465 | 0.6111 | 0.5828 |
| 4 | FCA_bin | 0.7100 | 0.6623 | 0.8611 | 0.5877 |
| 5 | Lazy_FCA | 0.7100 | 0.6491 | 0.9028 | 0.5991 |

Stroke prediction:

| | Model | Accuracy | Recall_0 | Recall_1 | F_score |
|---|---|---|---|---|---|
| 0 | LogisticRegression | 0.8233 | 0.9383 | 0.3333 | 0.4176 |
| 1 | RandomForest | 0.8167 | 0.9383 | 0.2982 | 0.3820 |
| 2 | KNeighbors | 0.8100 | 0.9053 | 0.4035 | 0.4466 |
| 3 | DecisionTree | 0.7567 | 0.8189 | 0.4912 | 0.4341 |
| 4 | FCA_bin | 0.8000 | 0.8272 | 0.6842 | 0.5652 |
| 5 | FCA_pat | 0.7133 | 0.7819 | 0.4211 | 0.3582 |

Employee prediction:

| | Model | Accuracy | Recall_0 | Recall_1 | F_score |
|---|---|---|---|---|---|
| 0 | LogisticRegression | 0.7100 | 0.8614 | 0.3980 | 0.4727 |
| 1 | RandomForest | 0.8267 | 0.8911 | 0.6939 | 0.7234 |
| 2 | KNeighbors | 0.7900 | 0.9010 | 0.5612 | 0.6358 |
| 3 | DecisionTree | 0.8233 | 0.8861 | 0.6939 | 0.7196 |

### 3) Lazy-FCA classification with binary attributes

For the use Lazy classification, I binarized every dataset and then perform Lazy Classification algorithm. I tuned hyperparameters: method ("standard", "standard-support", "ratio-support") and alpha (0.1, 0.5, 0.9). As the result we can see values of accuracy score.

**Income prediction:**

Best parameters:

Method = "standard", alpha = 0.1

```
Method: standard | Alpha: 0.1 | Accuracy: 0.71
Method: standard | Alpha: 0.5 | Accuracy: 0.5833
Method: standard | Alpha: 0.9 | Accuracy: 0.0767
Method: standard-support | Alpha: 0.1 | Accuracy: 0.6767
Method: standard-support | Alpha: 0.5 | Accuracy: 0.3667
Method: standard-support | Alpha: 0.9 | Accuracy: 0.22
Method: ratio-support | Alpha: 0.1 | Accuracy: 0.6033
Method: ratio-support | Alpha: 0.5 | Accuracy: 0.6233
Method: ratio-support | Alpha: 0.9 | Accuracy: 0.67
```

**Stroke prediction:**

```
Method: standard | Alpha: 0.1 | Accuracy: 0.7867
Method: standard | Alpha: 0.5 | Accuracy: 0.3467
Method: standard | Alpha: 0.9 | Accuracy: 0.0
Method: standard-support | Alpha: 0.1 | Accuracy: 0.7933
Method: standard-support | Alpha: 0.5 | Accuracy: 0.79
Method: standard-support | Alpha: 0.9 | Accuracy: 0.7867
Method: ratio-support | Alpha: 0.1 | Accuracy: 0.79
Method: ratio-support | Alpha: 0.5 | Accuracy: 0.8
Method: ratio-support | Alpha: 0.9 | Accuracy: 0.78
```

Best parameters:

Method = "ratio-support", alpha = 0.5

**Employee prediction:**

```
Method: standard | Alpha: 0.1 | Accuracy: 0.68
Method: standard | Alpha: 0.5 | Accuracy: 0.6933
Method: standard | Alpha: 0.9 | Accuracy: 0.2267
Method: standard-support | Alpha: 0.1 | Accuracy: 0.7167
Method: standard-support | Alpha: 0.5 | Accuracy: 0.71
Method: standard-support | Alpha: 0.9 | Accuracy: 0.6767
Method: ratio-support | Alpha: 0.1 | Accuracy: 0.7233
Method: ratio-support | Alpha: 0.5 | Accuracy: 0.7033
Method: ratio-support | Alpha: 0.9 | Accuracy: 0.7567
```

Best parameters:

Method = “ratio-support”, alpha = 0.9

### 4) Lazy-FCA with pattern structures

I performed 9 times Lazy Classification algorithm with pattern structured in the same way.

Final results are shown below:

**Income prediction:**

| | Model | Accuracy | Recall_0 | Recall_1 | F_score |
|---|---|---|---|---|---|
| 0 | LogisticRegression | 0.8167 | 0.9035 | 0.5417 | 0.5865 |
| 1 | RandomForest | 0.8333 | 0.8816 | 0.6806 | 0.6622 |
| 2 | KNeighbors | 0.8167 | 0.8553 | 0.6944 | 0.6452 |
| 3 | DecisionTree | 0.7900 | 0.8465 | 0.6111 | 0.5828 |
| 4 | FCA_bin | 0.7100 | 0.6623 | 0.8611 | 0.5877 |
| 5 | Lazy_FCA | 0.7100 | 0.6491 | 0.9028 | 0.5991 |

**Stroke prediction:**

| | Model | Accuracy | Recall_0 | Recall_1 | F_score |
|---|---|---|---|---|---|
| 0 | LogisticRegression | 0.8233 | 0.9383 | 0.3333 | 0.4176 |
| 1 | RandomForest | 0.8167 | 0.9383 | 0.2982 | 0.3820 |
| 2 | KNeighbors | 0.8100 | 0.9053 | 0.4035 | 0.4466 |
| 3 | DecisionTree | 0.7567 | 0.8189 | 0.4912 | 0.4341 |
| 4 | FCA_bin | 0.8000 | 0.8272 | 0.6842 | 0.5652 |
| 5 | FCA_pat | 0.7133 | 0.7819 | 0.4211 | 0.3582 |

**Employee prediction:**

| | Model | Accuracy | Recall_0 | Recall_1 | F_score |
|---|---|---|---|---|---|
| 0 | LogisticRegression | 0.7100 | 0.8614 | 0.3980 | 0.4727 |
| 1 | RandomForest | 0.8267 | 0.8911 | 0.6939 | 0.7234 |
| 2 | KNeighbors | 0.7900 | 0.9010 | 0.5612 | 0.6358 |
| 3 | DecisionTree | 0.8233 | 0.8861 | 0.6939 | 0.7196 |
| 4 | FCA_bin | 0.7567 | 0.8614 | 0.5408 | 0.5922 |
| 5 | FCA_pat | 0.7367 | 0.8168 | 0.5714 | 0.5864 |

## Conclusion

I applied the Lazy (FCA) algorithm to three datasets. In the overall evaluation, it falls in the middle of the scoreboard. However, it's worth noting that the pattern classifier tends to perform poorly when all attributes in the datasets are numerical.