**FEDERAL STATE AUTONOMOUS EDUCATIONAL**

**INSTITUTION FOR HIGHER EDUCATION**

**NATIONAL RESEARCH UNIVERSITY**

**HIGHER SCHOOL OF ECONOMICS**

**Faculty of Computer Science**

# BIG HOMEWORK REPORT

## On the course

## ORDERED SETS IN DATA ANALYSIS

## Lazy FCA

Student:

Tarasova Polina

Group: DS231

Supervisor:

D.Sc., Prof.

Kuznetsov Sergey Olegovich

Moscow 2023

## Introduction

In this paper will be introduced implementation of a Lazy FCA classification algorithm base on pattern structures. Proposed algorithm was compared with baseline Lazy FCA algorithm and popular models: Random Forest, Decision Tree, Logistic Regression, KNN and Naive Bayes. For comparison I used three datasets:

- **Heart Attack Analysis & Prediction Dataset**

  https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-prediction-dataset
- **Loan Defaults**

  https://www.kaggle.com/datasets/joebeachcapital/loan-default
- **Hotel Booking**

  https://www.kaggle.com/datasets/mojtaba142/hotel-booking

You can find all code in my GitHub repository: https://github.com/hellz-bellz/bhw-lazyfca

## Preprocessing and feature selection

All datasets were binarized: all numerical features were discretized, many valued categorical features were removed, then one-hot encoding technic was used.

## Training and metrics

The metrics for evaluation: accuracy, F1-macro and F1-binary. For training 10-fold Cross-Validation method was used.

Also, the Cross-Validation function was created for training LazyFCA models.

## Standard models

| | **Heart Attack** | | | **Loan Defaults** | | | **Hotel Booking** | | |
|---|---|---|---|---|---|---|---|---|---|
| | **ACC** | **F1-M** | **F1-B** | **ACC** | **F1-M** | **F1-B** | **ACC** | **F1-M** | **F1-B** |
| **DT** | 0.830 | 0.826 | 0.834 | **0.999** | **0.998** | **0.998** | 0.810 | 0.588 | 0.887 |
| **RF** | **0.954** | **0.934** | **0.899** | 0.992 | 0.990 | 0.986 | 0.850 | 0.541 | 0.915 |
| **LR** | 0.790 | 0.788 | 0.788 | 0.994 | 0.992 | 0.989 | 0.870 | 0.596 | 0.926 |
| **KNN** | 0.840 | 0.838 | 0.841 | 0.939 | 0.922 | 0.885 | **0.880** | **0.583** | **0.933** |
| **NB** | 0.825 | 0.824 | 0.826 | **0.999** | **0.998** | **0.998** | 0.850 | 0.598 | 0.914 |

## Lazy FCA

| | Heart Attack | | | Loan Defaults | | | Hotel Booking | | |
|---|---|---|---|---|---|---|---|---|---|
| | **ACC** | **F1-M** | **F1-B** | **ACC** | **F1-M** | **F1-B** | **ACC** | **F1-M** | **F1-B** |
| **BBC** | 0.525 | 0.423 | 0.522 | **0.995** | **0.994** | **0.991** | **0.780** | **0.595** | **0.861** |
| **PBC** | **0.740** | **0.716** | **0.752** | 0.739 | 0.483 | 0.601 | 0.750 | 0.613 | 0.823 |

## Conclusion

The datasets that I've chose all have different shape and target distribution. That and as well as the feature selection caused a great difference in results and training time.

The proposed algorithms has comparable results with popular classification models, but they has some problems with training time and overfitting, so there are a huge field for research and testing different approaches, especially in feature selection and preprocessing field.