Homework Report

Author: Kozlov Vasilii

Lazy FCA

Faculty of Computer Sciences
Higher School of Economics
Moscow

Contents

1 Data

I've chosen the following datasets:

• Heart Disease:

https://archive.ics.uci.edu/dataset/45/heart+disease

• Car Evaluation:

https://archive.ics.uci.edu/dataset/19/car+evaluation

• Bank Marketing:

https://archive.ics.uci.edu/dataset/222/bank+marketing

You can find all code in my GitHub repository https://github.com/Vaselyok/FCA_Big_Homework/blob/main/My_project.ipynb. In this repository you can find DataSets folder containing all used datasets separated by features \ target. Also this repository contains python notebook with code generating following results.

2 Classical methods used and parameter tuning

For all these datasets I used consecutively sklearn implementations of KNN, Logistic Regression and Random Forest algorithms.

I used sklearn GridSearchCV method for grid search of the best hyperparameters from the given parameter_grid set with values close by meaning to the task (for example: for KNN when the whole training set consists of 300 elements 200 neighbours will be most likely excessive).

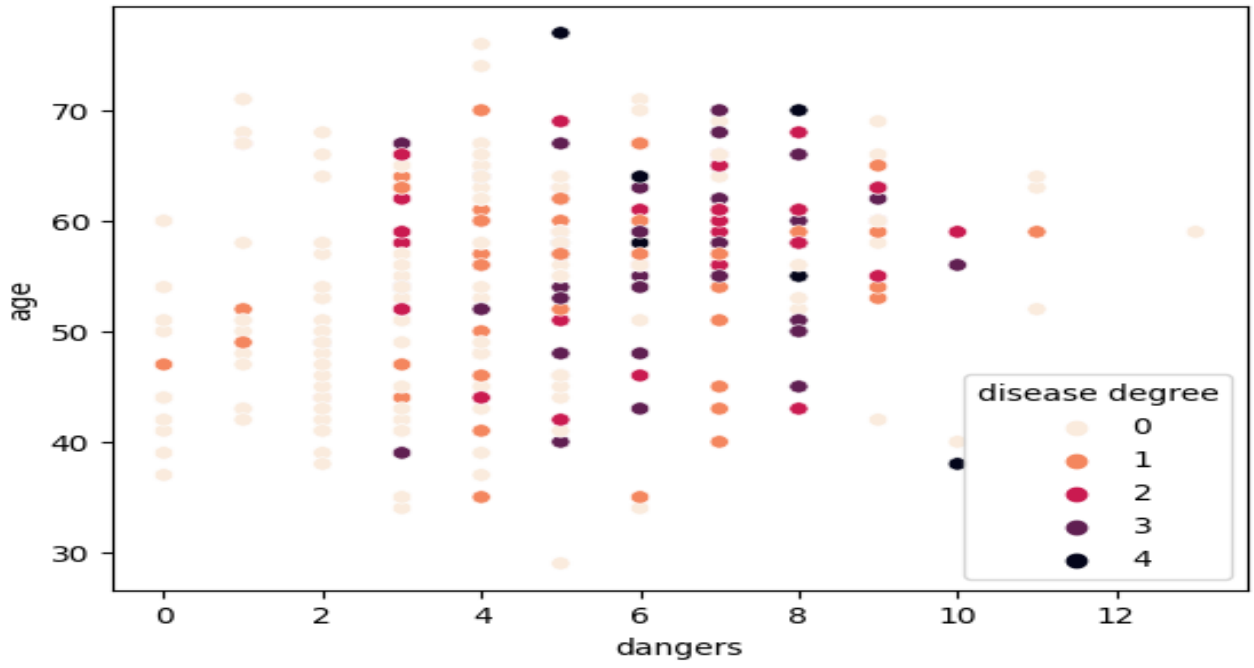
• Set of parameters for grid search for Random Forest :

parameter_grid = {'n_estimators', 'max_depth', 'max_leaf_nodes'}

• Set of parameters for grid search for KNN:

parameter_grid = {'n_neighbors'}

In the case of Heart Disease dataset additional feature 'dangers' was created.'dangers' feature represent weighted amount of parameters where significant deviations from the medically normal were observed. For example, when cp feature is equal to 1 it means the existence of a typical angina chest pain type which has a high correlation with the fact that this person has a heart disease, so +5 is added to 'dangers', on the other hand if thalac feature is high it means that high heart rate was achieved but this factor is much less correlated with the existence of a disease, so only +1 is added to the 'dangers'. As we can see this feature is quite informative:



3 FCA and model comparison

For FCA I used already created fcalc function. To use it on the data, I previously binarized the data. When there were few categories (<5) I used one hot encoding with pandas function get_dummies when there were numerical features, I split the data on <4 categories so that approximately the same number of samples will be in each. Target feature were mapped as follows:

• For Car Estimation I've just used get_dummies for the whole dataset as soon as there are only 6 features in this dataset.

• For Heart Disease dataset each feature was divided on groups according to medical information about the normal parameters

• Bank Subscription is the only dataset where exist some NaN values, so I replaced NaN with string Not given to create additional category which might have been useful. Also I mapped all features with binary values to (True,False) domain and all other categorical features with help of get_dummies to (True,False), so that FCA will be applicable.

When FCA was performed, I created a summary table with values of different metrics on these models:

DataSet	Method	recall	precision	accuracy	f1 score	ROC AUC
Bank Subscription	FCA	0.44	0.42	0.87	0.43	0.68
	KNN	0.18	0.56	0.89	0.27	0.58
	Logistic regression	0.35	0.64	0.90	0.45	0.66
	RandomForest	0.13	0.79	0.89	0.22	0.56
Cars	FCA	1.00	0.99	1.00	0.99	1.00
	KNN	0.86	0.99	0.95	0.92	0.93
	Logistic regression	0.93	0.93	0.96	0.93	0.95
	RandomForest	0.98	0.97	0.98	0.98	0.98
Heart Disease	FCA	0.85	0.83	0.86	0.84	0.87
	KNN	0.63	0.65	0.68	0.64	0.67
	Logistic regression	0.83	0.89	0.88	0.86	0.87
	RandomForest	0.80	0.87	0.86	0.84	0.85

4 Conclusion

We can see that FCA algorithm perform reasonably good. For Car Estimation FCA is absolutely the best algorithm by all metrics that were used. On the other hand, FCA algorithm was the slowest one and for large datasets(for bigger ones it might take even longer) it takes several hours to train a model when classical algorithms that were chosen by me for this work learned in less than 20 minutes each.