Prediction of Heart Attacks: Using Lazy Learning and Formal Concept Analysis

Tatiana Iakovleva

December 13, 2022

1. **Background.**

   This study uses "Heart Disease Dataset" from UCI Machine Learning Repository. This dataset contains 76 attributes, but publishers refer to use only 14 of them. The purpose of the study is to analyze factors that explain the presence or the absence of heart attack using Lazy Learning and Formal Concept Analysis and compare these results with other models.

   Heart disease is one of the biggest issues of mortality among the population of the world. Prediction of this disease can lead to earlier detection of the likelihood of a problem. And in this case, it will help a large number of people to extend their lives.

2. **Dataset Description.**

   1. Age: the age of the individual.

   2. Sex: the gender of the individual using the following format :

   1 = male

   0 = female

   3. Chest-pain type ("cp"): the type of chest-pain experienced by the individual using the following format :

   1 = typical angina

   2 = atypical angina

   3 = non — anginal pain

   4 = asymptotic

   4. Resting Blood Pressure("trestbps"): the resting blood pressure value of an individual in mmHg (unit).

   5. Serum Cholestrol ("chol"): the serum cholesterol in mg/dl (unit)

   6. Fasting Blood Sugar("fbs"): the fasting blood sugar value of an individual with 120mg/dl.

   If fasting blood sugar is higher than 120mg/dl: 1 (true)

   else : 0 (false)

   7. Resting ECG("restecg"): resting electrocardiographic results.

   0 = normal

   1 = having ST-T wave abnormality

   2 = left ventricular hyperthrophy

   8. Max heart rate achieved: the max heart rate achieved by an individual.

   9. Exercise induced angina:

   1 = yes

   0 = no

   10. ST depression induced by exercise relative to rest

   11.Peak exercise ST segment:

   1 = upsloping

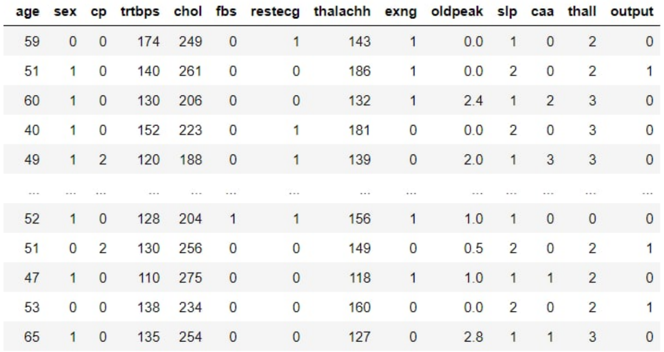
   2 = flat

   3 = downsloping

   12.Number of major vessels (0–3) colored by flourosopy

   13.Thal: the thalassemia.

   3 = normal



age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa	thall	output
59	0	0	174	249	0	1	143	1	0.0	1	0	2	0
51	1	0	140	261	0	0	186	1	0.0	2	0	2	1
60	1	0	130	206	0	0	132	1	2.4	1	2	3	0
40	1	0	152	223	0	1	181	0	0.0	2	0	3	0
49	1	2	120	188	0	1	139	0	2.0	1	3	3	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
52	1	0	128	204	1	1	156	1	1.0	1	0	0	0
51	0	2	130	256	0	0	149	0	0.5	2	0	2	1
47	1	0	110	275	0	0	118	1	1.0	1	1	2	0
53	0	0	138	234	0	0	160	0	0.0	2	0	2	1
65	1	0	135	254	0	0	127	0	2.8	1	1	3	0

6 = fixed defect

7 = reversible defect

14.Diagnosis of heart disease: if the individual is suffering from heart disease or not:

0 = absence

1 = presence

3. **Data Preparation.**

In the dataset there were 3 main types of data that were processed differently:

1. Binary (for example, sex, presence of disease). For such features, the values were replaced with 0 and 1. The specific value for which we need to specify 0 (or 1) does not play a role, so it was chosen randomly.

2. Categorical (for example, the type of pain). For each such feature, new ones were added, one for each value. The value 1 was set if the given attribute had this value. After that, the original categorical features were removed.

3. Numeric (for example, blood pressure). For each such feature there were three options: in FCA algorithms instead of continuous variables indicator of smaller values than average ones were used. Moreover, then each variable was devided into 4 equaled in terms of the size groups. During other models' preparation Standard Scaler was implemented.

4. **Fca Algorithms.**

1. Baseline algorithm called "Generators framework".

It is necessary to count the number of counterexamples for positive examples. For each positive example $x_{pos} \in X_{pos}$ we compute the intersection $x \cap x_{pos}$. Then, we count the counterexamples for this intersection, that is the number of negative examples $x_{neg} \in X_{neg}$ containing intersection;

Dually, count the number of counterexamples for negative examples. For each negative example $x_{neg} \in X_{neg}$ we compute the intersection $x \cap x_{neg}$. Then, we count the counterexamples for obtained intersection, that is the number of positive examples $x_{pos} \in X_{pos}$ containing intersection $x \cap x_{neg}$.

2. Lazy Classification with Pattern Structures.

Pattern structures used here: for each row in testing set for the nominal attributes, there was the set theoretic intersection. All numeric data was converted to categorical data. Based on this intersection similar examples from the train data was defined, so extension was calculated. Then, the target of these observations from the extension were used in order to classify the examples.

5. **Quality of the Models.**

In order to determine the quality of the models some basic metrics were calculated:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

Table 1: Results of the FCA Classification.

Algorithm	Accuracy	Precision	Recall
Baseline (average)	0.813	0.827	0.832
Pattern Structures (average)	0.593	0.909	0.217
Baseline (4 groups)	0.828	0.815	0.886
Pattern Structures (4 groups)	0.648	0.6	0.99

6. **Results of the FCA Classification.**

   The target is balanced, that is why Accuracy score can be used as the main criteria.

7. **Results of other models Classification.**

Table 2: Results of other models Classification.

Algorithm	Accuracy	Precision	Recall
SVM	0.846	0.796	0.935
Logistic Regression	0.868	0.827	0.935
Decision Tree	0.835	0.830	0.848
Xgboost	0.813	0.814	0.813
Catboost Classification	0.868	0.815	0.957

8. **Conclusion.**

   To sum up, all models do not have too much differences in terms of accuracy, however, it is obvious that catboost classification is best in terms of presented metrics. In my opinion, lazy classification models also showed good results but they can be improved in the future research.