Report

Big Home Work. Lazy Formal Concept Analysis

Usmanova K.

15.12.2019

1. Background

There was used “Heart Disease Data set” from UCI Machine Learning Repository (https://www.kaggle.com/ronitf/heart-disease-uci). This dataset contains 76 attributes, but publishers refer to use only 14 of them. The goal of this research is to identify the presence of heart disease in the patients. Heart disease is one of the biggest causes of morbidity and mortality among the population of the world. Prediction of cardiovascular disease is regarded as one of the most important subjects in the section of clinical data analysis. In virtue of this “Heart Diseas” dataset we can find the range of conditions that affect your heart.

2. Description of dataset

The dataset consists of 303 rows and 14 columns. As it was mentioned before, the main purpose of the dataset is to determine the presence of heart disease (represented as a “target” value in dataset) in patients. Target value is integer valued from 0 (no presence of disease) and 1 (its presence). There are following features:

1. *Age*: displays the age of the individual.

2. *Sex*: displays the gender of the individual using the following format :
   1 = male
   0 = female

3. *Chest-pain type (“cp”)*: displays 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”)*: displays the resting blood pressure value of an individual in mmHg (unit)

5. *Serum Cholestrol(“chol”)*: displays the serum cholesterol in mg/dl (unit)

6. *Fasting Blood Sugar("fbs")*: compares the fasting blood sugar value of an individual with 120mg/dl.
   If fasting blood sugar > 120mg/dl then : 1 (true)
   else : 0 (false)

7. *Resting ECG("restecg")* : displays resting electrocardiographic results
   0 = normal
   1 = having ST-T wave abnormality
   2 = left ventricular hyperthrophy

8. *Max heart rate achieved* : displays 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*: displays the value which is an integer or float.

11. *Peak exercise ST segment* :
   1 = upsloping
   2 = flat
   3 = downsloping

12. *Number of major vessels (0–3) colored by flourosopy* : displays the value as integer or float.

13. *Thal* : displays the thalassemia :
   3 = normal
   6 = fixed defect
   7 = reversible defect

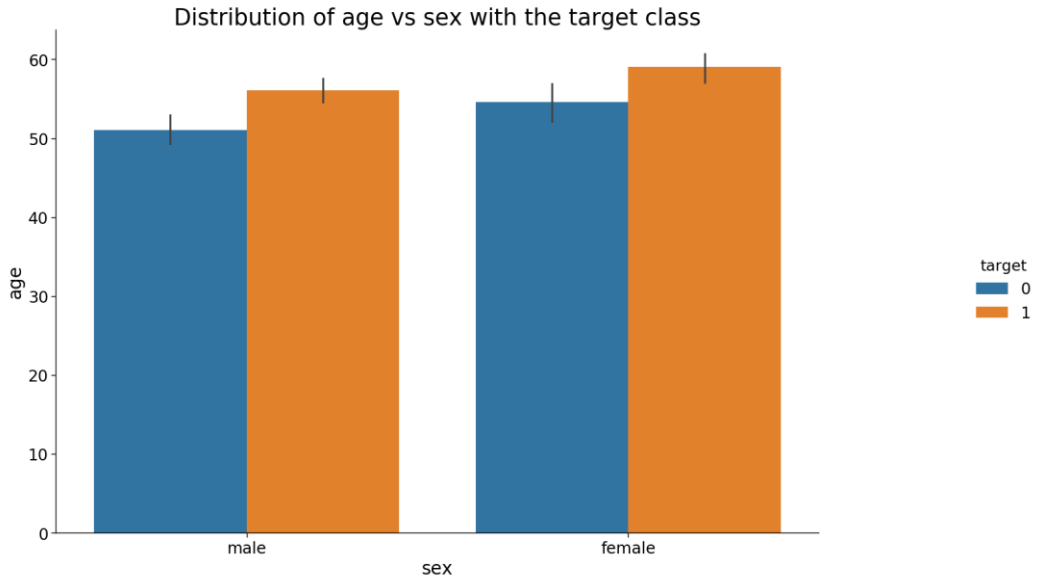
14. *Diagnosis of heart disease* : Displays whether the individual is suffering from heart disease or not :
   0 = absence
   1 = present

3. Data analysis

Let's take a look at the people's age who are suffering from the disease or not. So, target = 1 means that person is suffering from heart disease and target = 0 means the person is not suffering.

pic. 1

It is clear to see, that most people who are suffering are of the age of 58, followed by 57. We can conclude that people belonging to the age group 50+ are suffering from the disease (pic. 1).



pic. 2

Let us look at the distribution of age and gender for each target class (pic.2). We could say that for females who are suffering from the disease are older than males.

4. Data Pre-Processing

The dataset is shown below:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3	0
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3	0
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3	0
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3	0
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2	0

303 rows × 14 columns

First of all, we should binarize it (details in code):

	age29_40	age41_60	age61_77	male	female	cp0	cp1	cp2	cp3	trestbps94_120	...	ca0	ca1	ca2	ca3	ca4	thal0	thal1	thal2	thal3	target
0	0	0	1	1	0	0	0	0	1	0	...	1	0	0	0	0	0	1	0	0	1
1	1	0	0	1	0	0	0	1	0	0	...	1	0	0	0	0	0	0	1	0	1
2	0	1	0	0	1	0	1	0	0	0	...	1	0	0	0	0	0	0	1	0	1
3	0	1	0	1	0	0	1	0	0	1	...	1	0	0	0	0	0	0	1	0	1
4	0	1	0	0	1	1	0	0	0	1	...	1	0	0	0	0	0	0	1	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
298	0	1	0	0	1	1	0	0	0	0	...	1	0	0	0	0	0	0	0	1	0
299	0	1	0	1	0	0	0	0	1	1	...	1	0	0	0	0	0	0	0	1	0
300	0	0	1	1	0	1	0	0	0	0	...	0	0	1	0	0	0	0	0	1	0
301	0	1	0	1	0	1	0	0	0	0	...	0	1	0	0	0	0	0	0	1	0
302	0	1	0	0	1	0	1	0	0	0	...	0	1	0	0	0	0	0	1	0	0

I divided the data in the test and train set. Deviation of data was into an 70:30 ratio. It means that training size is 0.7 and test size is 0.3.

5. Lazy FCA classification algorithm

In this classification algorithm, we're doing following steps:

1) The dataset was divided into positive and negative classes. After, the algorithm try to determine the objects from unclassified set to the positive or negative classes.
2) The algorithm makes intents (positive and negative objects).
3) Then every unclassified object obtains its intent.
4) We intersect every unclassified object with positive ones, then the same step we do with negative objects.

5) There is following condition: if unclassified object has intersection only with positive intents then we define it as positive, if unclassified object has intersection only with negative intents then we define it as negative, otherwise there is contradiction.

Algorithm

Let's see the following definitions:
g - unclassified object
G_+ - set of positive training objects
G_- - set of positive training objects
g' – features of unclassified objects
g^+ - features of objects from positive set
g^- - features of objects from negative set

Metrics

To evaluate the quality of classification of test samples, I used following metrics:

- $\text{accuracy} = \frac{TP+TN}{N}$
- $\text{precision} = \frac{TP}{TP+FP}$
- $\text{recall} = \frac{TP}{TP+FN}$

Where N = TP + TN + FP + FN.

Aggregation function 1

I have added aggregation function to increase accuracy of classification:

$$\text{Pos} = \frac{1}{|G_+|}\sum_{g_i \epsilon G_+} |g' \cap g_i^+|$$

$$\text{Neg} = \frac{1}{|G_-|}\sum_{g_i \epsilon G_-} |g' \cap g_i^-|$$

This condition means that object will be classified as "positive" if max(Pos, Neg), otherwise it is "negative". Here, I have got 54% accuracy. There is following results:

```
Aggregation function #1:

accuracy:   54.94505494505495
precission:   54.94505494505495
recall:    100.0
```

This accuracy still needs enhancements.

Aggregation function 2

I modified my aggregation function to:

$$\text{Pos} = \frac{1}{|G_+|}\sum_{g_{i\epsilon G_+}} |g' \cap g_i^+|$$

$$\text{Neg} = \frac{1}{|G_-|}\sum_{g_{i\epsilon G_-}} |g' \cap g_i^-|$$

This condition means that object will be classified as “positive” if Pos>=Neg, otherwise it is “negative”. Accuracy of classification significantly increased till 80%. The results:

```
Aggregation function #2:

accuracy:   80.21978021978022
precission:   79.62962962962963
recall:    86.0
```

Comparison with classical classification algorithms

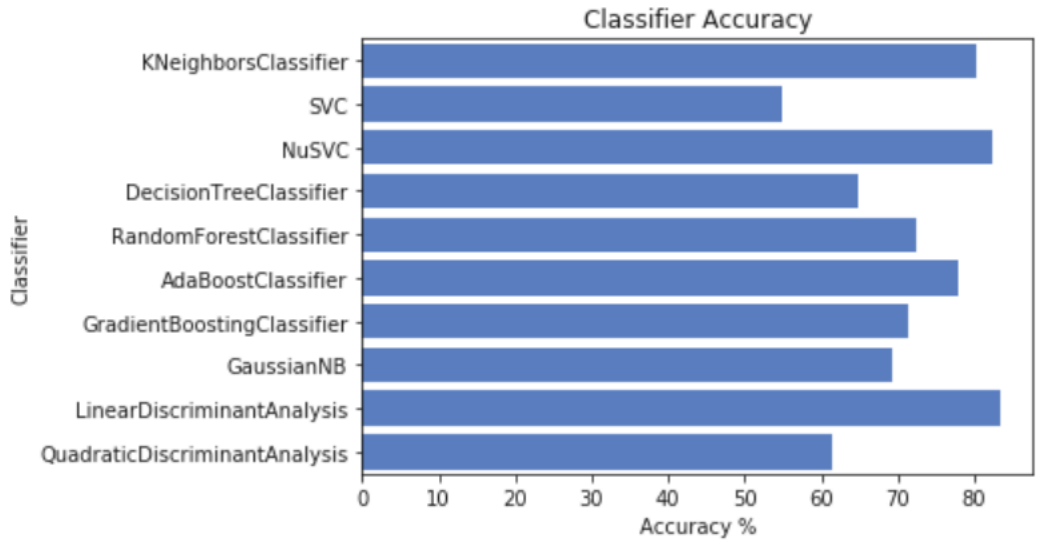
Here, I decided to compare the results with the other classification algorithms. There were used next classifiers:

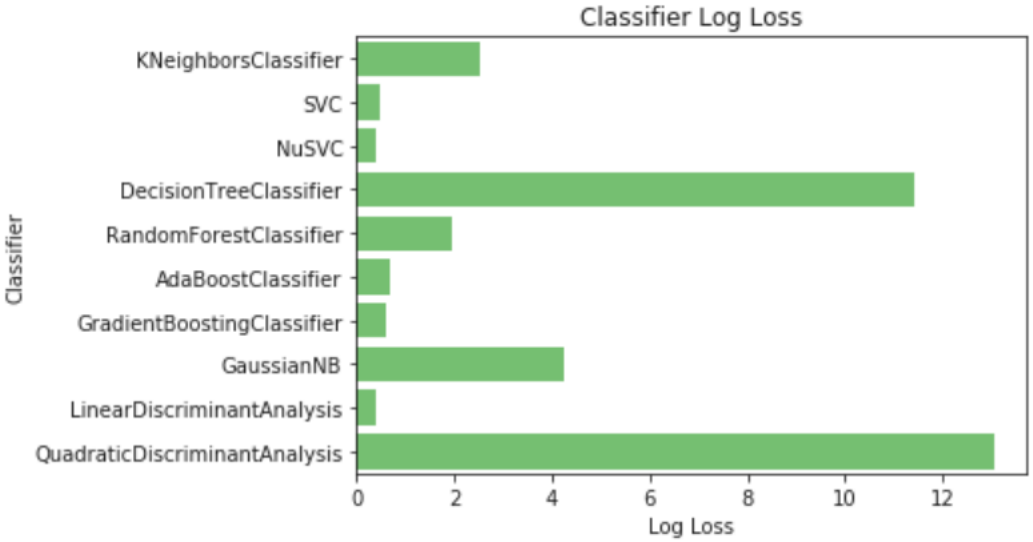
1) K - nearest neighbors algorithm (KNN)
2) Support vector classification(SVC)
3) NuSVC
4) Decision Tree Classifier
5) Random Forest Classifier
6) Ada Boosting Classifier
7) Gradient Boosting Classifier
8) Gaussian Naïve Bayes
9) Linear Discriminant Analysis
10) Quadratic Discriminant Analysis

Classifier	Accuracy	Precision	Recall	LogLoss core
KNN	80.2198%	82.0000%	82.0000%	2.542
SVC	54.9451%	54.9451%	100.0000%	0.453
NuSVC	82.4176%	84.0000%	84.0000%	0.381
DecisionTree	64.8352%	71.4286%	60.0000%	11.401
RandomForest	72.5275%	76.5957%	72.0000%	1.966
AdaBoost	78.0220%	78.8462%	82.0000%	0.668
GradientBoosting	71.4286%	75.0000%	72.0000%	0.61

GaussianNaiveBayes	69.2308%	77.5000%	62.0000%	4.253
LinearDiscriminant	83.5165%	84.3137%	86.0000%	0.398
QuadraticDiscriminant	61.5385%	82.6087%	38.0000%	13.073

According to given results, we could say that NuSVC and LinearDiscriminant have a bit higher accuracy in 2% and 3, while KNN has the same accuracy as LazyFCA result, the lowest result was provided by SVC classifier. It is clear to see that LazyFCA provides with pretty good accuracy than the other classifiers.





Logarithmic loss (related to cross-entropy) measures the performance of a classification model where the prediction input is a probability value between 0 and 1. The goal of our machine learning models is to minimize this value. A perfect model would have a log loss of 0. Log loss increases as the predicted probability diverges from the actual label. So predicting a probability of .012 when the actual observation label is 1 would be bad and result in a high log loss. There is a more detailed explanation of the justifications and math behind log loss.

Conclusion

The LazyFCA works pretty, but still needs some enhancements in aggregation function, because it depends on different points like binarization of objects, and finding the best condition for classification.

Code

There are two codes:

- “LazFCA1.ipynb” contains 1st aggregation function
- “LazFCA2.ipynb” contains 2nd aggregation function

https://github.com/usmk/Lazy-FCA-classifier