Data

Desctiption of dataset

This dataset contains the medical records of 299 patients who had heart failure, collected during their follow-up period, where each patient profile has 13 clinical features.

Attribute info

- age: age of the patient (years)
- anaemia: decrease of red blood cells or hemoglobin (boolean)
- high blood pressure: if the patient has hypertension (boolean)
- creatinine phosphokinase (CPK): level of the CPK enzyme in the blood (mcg/L)
- diabetes: if the patient has diabetes (boolean)
- ejection fraction: percentage of blood leaving the heart at each contraction (percentage)
- platelets: platelets in the blood (kiloplatelets/mL)
- sex: woman or man (binary)
- serum creatinine: level of serum creatinine in the blood (mg/dL)
- serum sodium: level of serum sodium in the blood (mEq/L)
- smoking: if the patient smokes or not (boolean)
- time: follow-up period (days)
- (target) death event: if the patient deceased during the follow-up period (boolean)

```
df = pd.read_csv('heart_failure_clinical_records_dataset.csv')
df.dropna()
df.head()
```

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time
0	75.0	0	582	0	20	1	265000.00	1.9	130	1	0	4
1	55.0	0	7861	0	38	0	263358.03	1.1	136	1	0	6
2	65.0	0	146	0	20	0	162000.00	1.3	129	1	1	7
3	50.0	1	111	0	20	0	210000.00	1.9	137	1	0	7
4	65.0	1	160	1	20	0	327000.00	2.7	116	0	0	8

For FCA classification approach data should be binary, so we binarized non-binary features. Numerical features were uniformly distributed across 7 equally spaced bins.

```
df
```

	anaemia	diabetes	high_blood_pressure	sex	smoking	DEATH_EVENT	age_0	age_1	age_2	age_3	...	platelets_4	platelets_5	platelets_6	creatinine_
0	0	0	1	1	0	1	1	1	1	1	...	0	0	0	
1	0	0	0	1	0	1	1	1	0	0	...	0	0	0	
2	0	0	0	1	1	1	1	1	1	0	...	0	0	0	
3	1	0	0	1	0	1	1	0	0	0	...	0	0	0	
4	1	1	0	0	0	1	1	1	1	0	...	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
294	0	1	1	1	1	0	1	1	1	0	...	0	0	0	
295	0	0	0	0	0	0	1	1	0	0	...	0	0	0	
296	0	1	0	0	0	0	0	0	0	0	...	1	1	1	
297	0	0	0	1	1	0	0	0	0	0	...	0	0	0	
298	0	0	0	1	1	0	1	0	0	0	...	0	0	0	

299 rows × 55 columns

Lazy FCA algorithm

Since the traditional algorithm is quite slow, we consider consider a variant of the voting algorithm for matching the plus of the context and the classified object (the same for the minus). This algorithm consists of the following steps:

1. We divide our dataset into positive and negative classes.
2. Our alorithm is trying to compare the classified object with each example from positive class.
3. More precisely, we compare $\sum |g' \cap g'_+|$ with $|g'| \cdot$ threshold.
4. Analogously, we compare $\sum |g' \cap g'_-|$ with $|g'| \cdot$ threshold. Here g' -features of unclassified objects, g_+, g_- - features of objects from positive/negative set.
5. If the sum of intersections is greater, then this example(from positive or negative class) votes for the classified object.
6. Then we normalize the sums of votes in both classes. And the normalized ones are used for classification.

Baseline accuracy of this algorithm with default threshold (0.1) is 0.67. To maximize accuracy we searched for the best value of thrashold over. We found that the $threshold = 0.8$ miximize the accuracy score for our dataset and the accuracy incresed to 0.83.

Metrics

To evaluate this algorithm, 13 metrics were used:

- True positive
- True Negative
- False Positive
- False Negative
- True Positive Rate
- True Negative Rate
- Negative Predictive Value
- False Positive Rate
- False Discovery Rate
- Accuracy
- Precision
- Recall
- F1 score

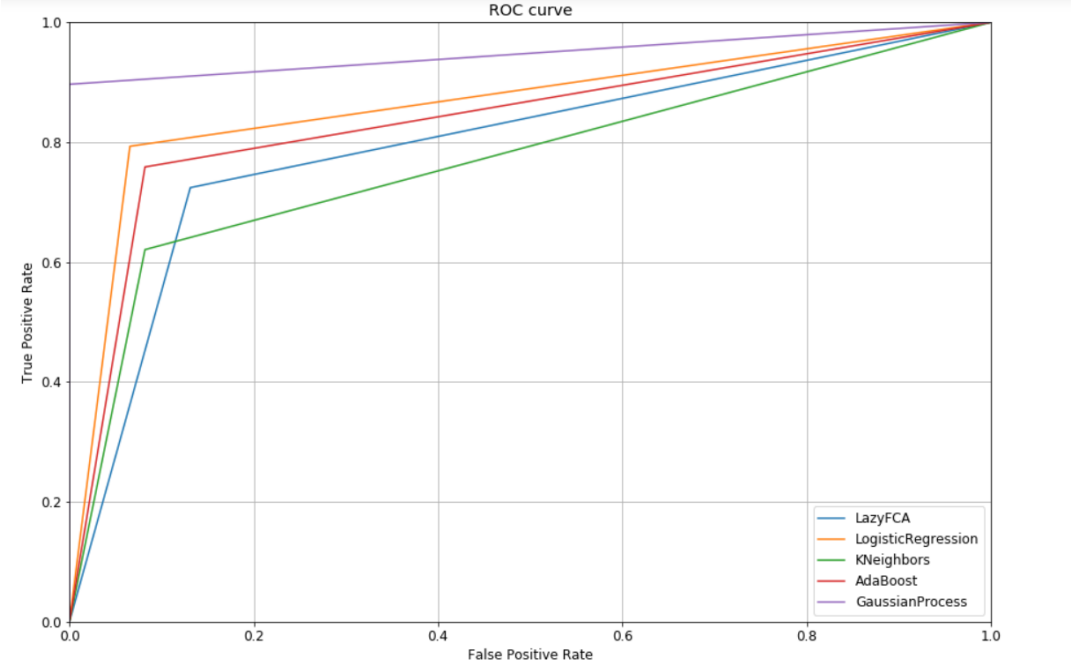
where accuracy, precision, recall and f1 measure were taken from sklearn library.

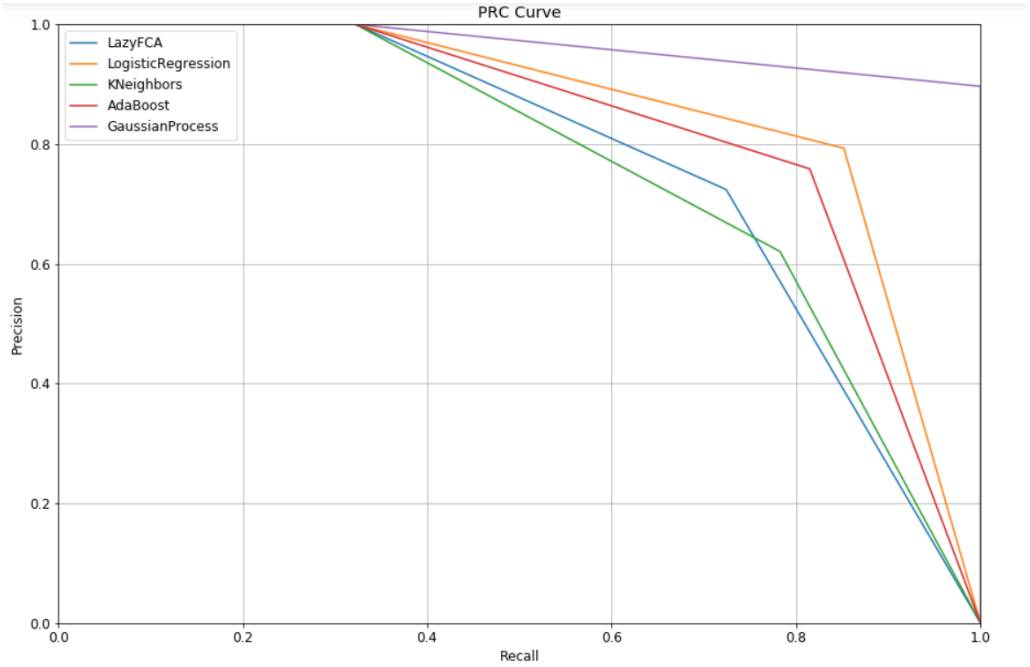
Comparison with other alorithms

We use previously presented metrics to compare our agorithm with 4 other popular algorithms, namely LogisticRegression, KNN(with $k = 5$), AdaBoost and GaussianProcess. The results are displayed in the following table:

Metric	LazyFCA	LogisticRegression	KNN	AdaBoost	GaussianProcess
True positive	21	23	18	22	26
True Negative	54	57	56	56	61
False Positive	7	4	5	5	0
False Negative	8	6	1	7	3
True Positive Rate	0.724	0.793	0.62	0.758	0.896
True Negative Rate	0.88	0.934	0.91	0.918	1
Negative Predictive Value	0.87	0.904	0.835	0.88	0.95
False Positive Rate	0.114	0.065	0.08	0.08	0
False Discovery Rate	0.25	0.148	0.217	0.185	0
Accuracy	0.833	0.88	0.82	0.86	0.96
Precision	0.75	0.85	0.78	0.81	1
Recall	0.724	0.79	0.62	0.76	0.896
F1	0.736	0.82	0.69	0.785	0.945

We plot Roc and PRC curves to visualise results:





Conclusion

As we can see from the table and graphs above, LazyFCA performs slightly better than KNN and at the same time slightly worse than the others. Apart from that, with attuned thershold parameter, our algorithms shows good results but still needs some improvements. The correct categorization and further binarization of data play an important role here. Also, lazy classification algorithms have significant "customization" capabilities, due to which they can theoretically surpass the standard ones in quality.