

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МО ЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №1**  
**по дисциплине «Машинное обучение»**  
**Тема: Предобработка данных**

Студент гр. 6304

Иванов Д.В.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2020

## Цель работы

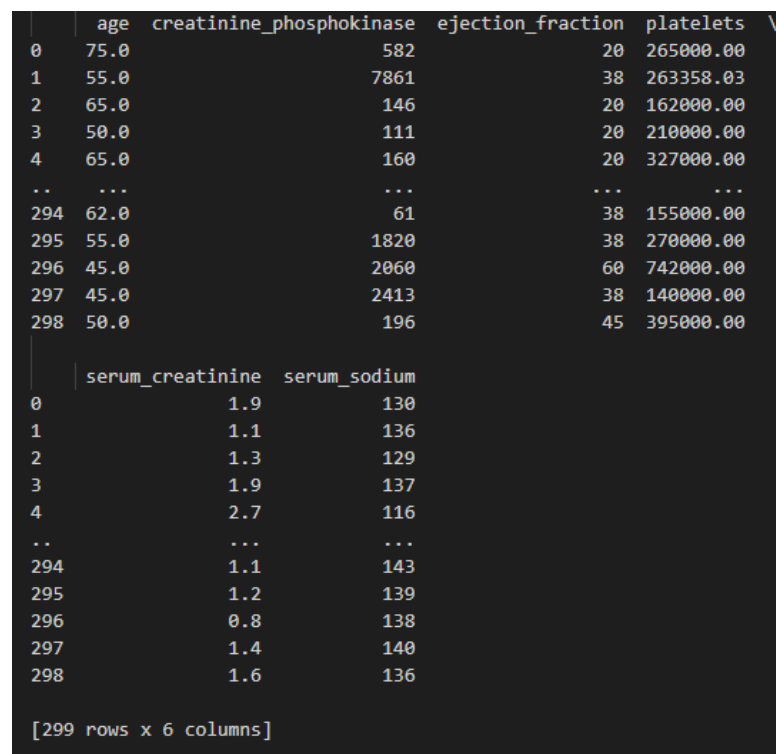
Ознакомиться с методами предобработки данных из библиотеки Scikit Learn

## Ход работы

### Загрузка данных

1. В датафрейм загружен исходный датасет, исключены бинарные признаки и признаки времени (рис. 1).

```
df = pd.read_csv('heart_failure_clinical_records_dataset.csv')
df = df.drop(columns=['anaemia', 'diabetes', 'high_blood_pressure', 'sex', 'smoking', 'time', 'DEATH_EVENT'])
print(df)
```



	age	creatinine_phosphokinase	ejection_fraction	platelets	\
0	75.0	582	20	265000.00	
1	55.0	7861	38	263358.03	
2	65.0	146	20	162000.00	
3	50.0	111	20	210000.00	
4	65.0	160	20	327000.00	
..	...	...	...	...	...
294	62.0	61	38	155000.00	
295	55.0	1820	38	270000.00	
296	45.0	2060	60	742000.00	
297	45.0	2413	38	140000.00	
298	50.0	196	45	395000.00	

	serum_creatinine	serum_sodium
0	1.9	130
1	1.1	136
2	1.3	129
3	1.9	137
4	2.7	116
..	...	...
294	1.1	143
295	1.2	139
296	0.8	138
297	1.4	140
298	1.6	136

[299 rows x 6 columns]

Рис. 1 — Загруженный датасет

2. Построены гистограммы признаков (рис. 2).

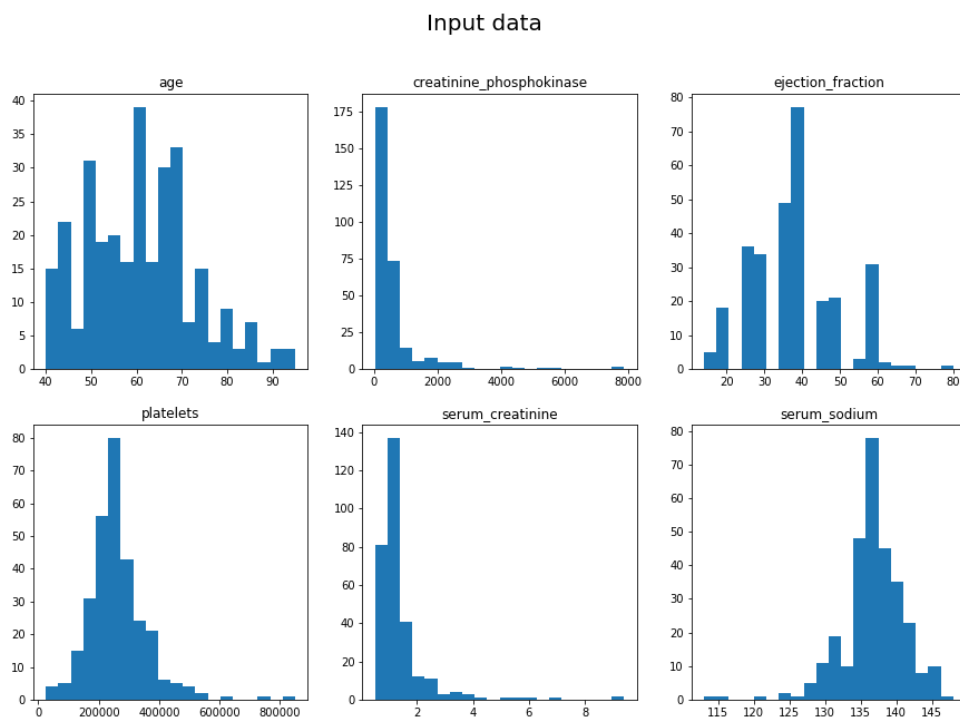


Рисунок 2 — Гистограмма признаков

3. На основании гистограмм определены диапазоны значений каждого признака и значения, около которых лежит наибольшее количество наблюдений.

Признак	Диапазон	Значение с наибольшим количеством наблюдений
age	(40, 95)	61.25
creatinine_phosphokinase	(0, 7900)	200
ejection_fraction	(7, 80)	38.3
platelets	(40000, 840000)	260000
serum_creatinine	(0.25, 9.5)	1.4
serum_sodium	(110, 150)	137.5

4. Датафрейм приведен к формату numpy.

## Стандартизация данных

1. Выполнена стандартизация всех наблюдений на основе первых 150, затем построены гистограммы признаков (рис. 3)

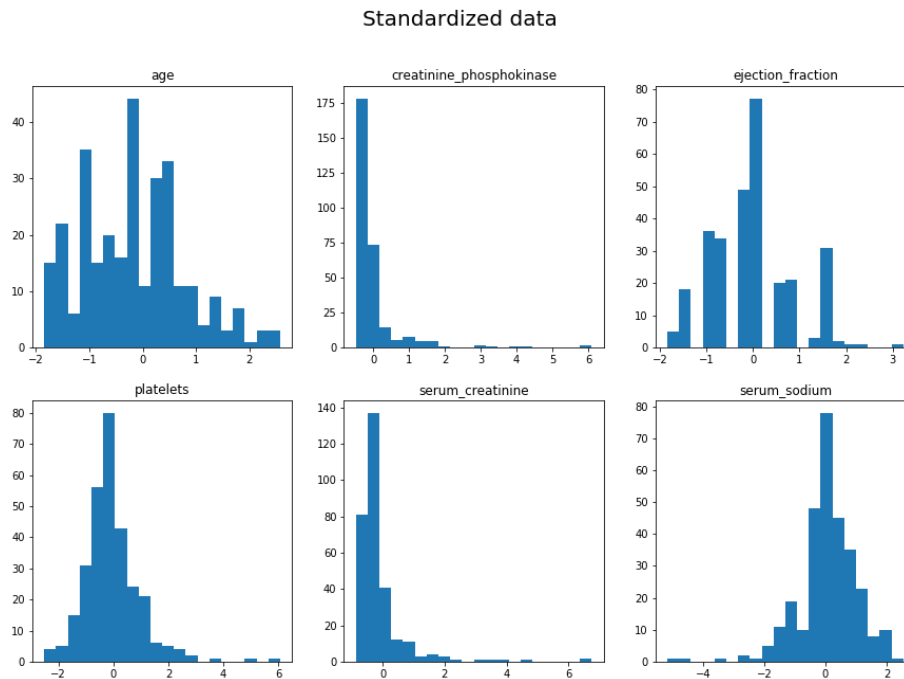


Рисунок 3 — Гистограмма стандартизированных признаков

2. На основании гистограмм были определены диапазоны значений каждого из признаков, а также возле какого значения лежит наибольшее количество наблюдений.

Признак	Диапазон	Значение с наибольшим количеством наблюдений
age	(-2, 2.6)	-0.1
creatinine_phosphokinase	(-0.7, 6.4)	-0.1
ejection_fraction	(-2, 3.3)	0
platelets	(-3, 6.5)	0
serum_creatinine	(-1.5, 7)	-0.2
serum_sodium	(-5.5, 2.3)	0

Из-за примененного преобразования (стандартизации) диапазон и значение с наибольшим количеством наблюдений полученных данных изменились.

3. Проведена стандартизация на полном наборе наблюдений.

```
full_scaler = preprocessing.StandardScaler()
full_data_scaled = full_scaler.fit_transform(data)
```

4. Вычислены мат. ожидание и СКО каждой выборки.

Выборка	Статистика	age	creatinine_phosphokinase	ejection_fraction	platelets	serum_creatinine	serum_sodium
Оригинальная	Мат. ожид.	60.833	581.839	38.084	263358.029	1.394	136.625
	СКО	11.875	968.664	11.815	97640.548	1.032	4.405
Стандартизованная на 150	Мат. ожид.	-0.170	-0.021	0.011	-0.035	-0.109	0.038
	СКО	0.955	0.816	0.908	1.017	0.887	0.972
Стандартизованная	Мат. ожид.	5.703e-16	0.0	-3.267e-17	7.723e-17	1.426e-16	-8.675e-16
	СКО	1.0	1.0	1.0	1.0	1.0	1.0

На основании результатов сравнения можно сделать вывод, что стандартизация имеет следующую форму:

$$Y = \frac{X - \mu(X)}{std(X)}, \text{ где } \mu(X) - \text{мат. ожидание, а } std(X) - \text{СКО.}$$

5. Поля *mean\_* и *var\_* объекта *StandardScaler* содержат мат. ожидание и дисперсия величин, на основании которых стандартизируются данные.

## Приведение к диапазону

1. Посредством *MinMaxScaler* данные приведены к диапазону (рис. 4)

```
min_max_scaler = preprocessing.MinMaxScaler().fit(data)
data_min_max_scaled = min_max_scaler.transform(data)
```

MinMaxScaled data

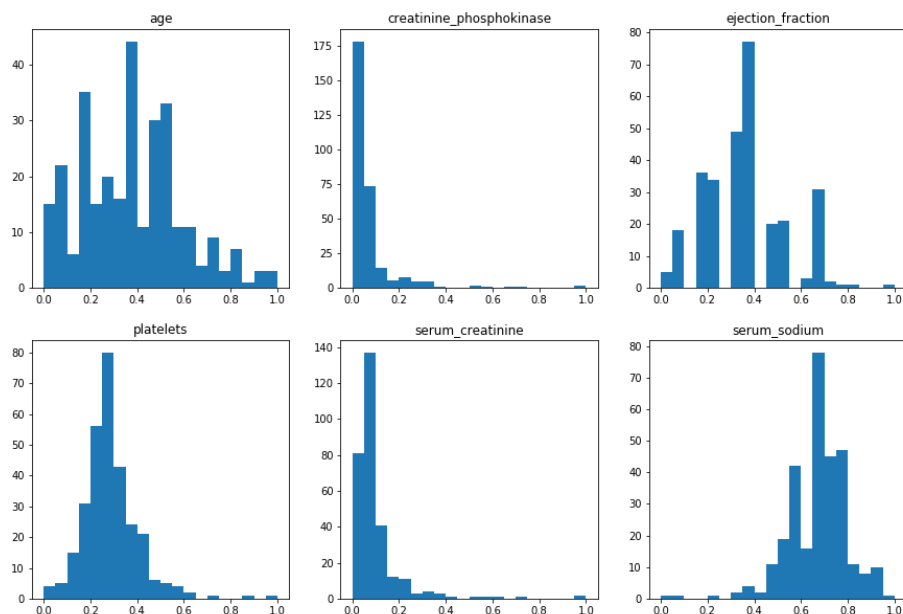


Рисунок 4 — Гистограмма после MinMaxScaler

Основываясь на гистограммах, можно заметить, что данные приводятся к диапазону  $[0,1]$ . Подобное преобразование можно осуществить с помощью формулы:

$$Y = \frac{X - \min(X)}{\max(X) - \min(X)}$$

2. Определены минимальное и максимальное значения каждого признака, посредством объекта *MinMaxScaler*.

	age	creatinine_phosphokinase	ejection_fraction	platelets	serum_creatinine	serum_sodium
Min	4.00e+01	2.30e+01	1.40e+01	2.51e+04	5.00e-01	1.13e+02
Max	9.500e+01	7.861e+03	8.000e+01	8.500e+05	9.400e+00	1.480e+02

3. С помощью *MaxAbsScaler* и *RobustScaler* выполнено приведение данных к диапазону (рис. 5 - 6)

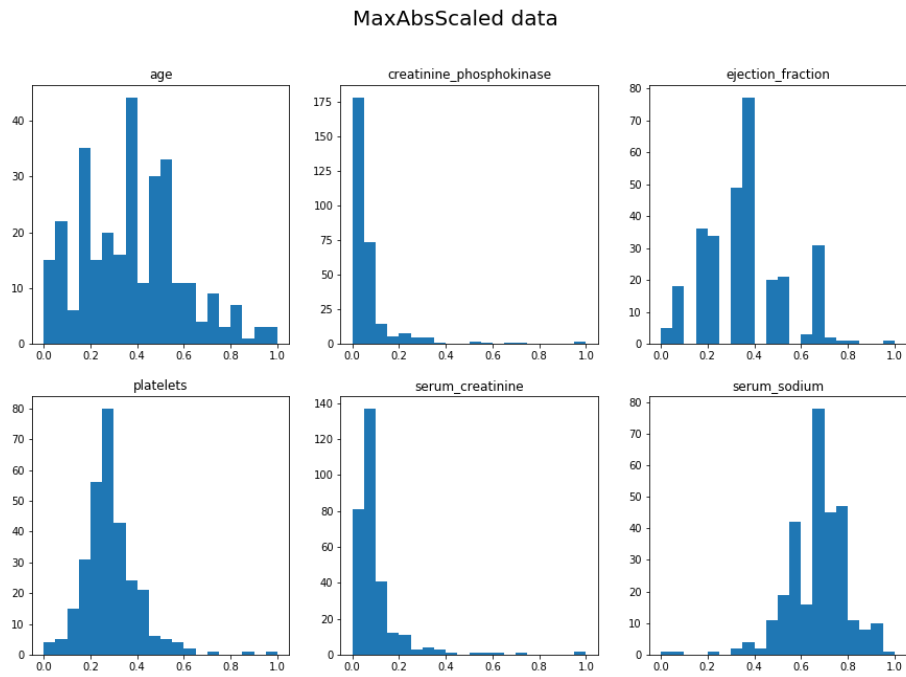


Рисунок 5 — Гистограмма после MaxAbsScaler

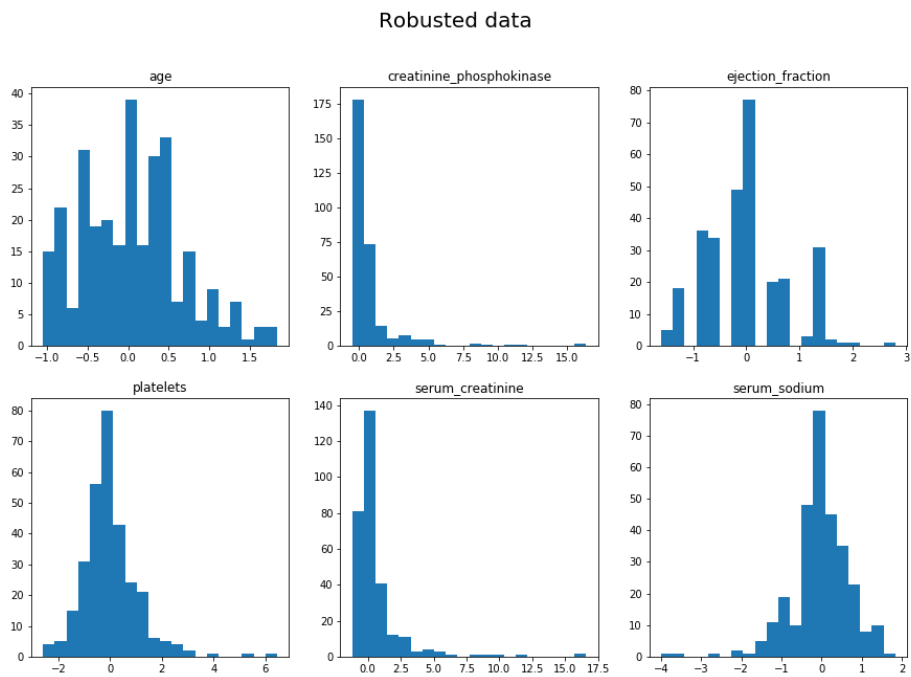


Рисунок 6 — Гистограмма после RobustScaler

*MaxAbsScaler* приводит данные таким образом, что максимальное по модулю значение равно 1.

*RobustScaler* вычитает медиану и масштабирует данные в соответствии с межквартильным размахом.

4. Также была написана функция, которая приводит данные к диапазону  $[-5, 10]$ .

```
def range_5_10(data): return preprocessing.MinMaxScaler().fit(data).transform(data)*15-5
```

Результат на рис. 7.

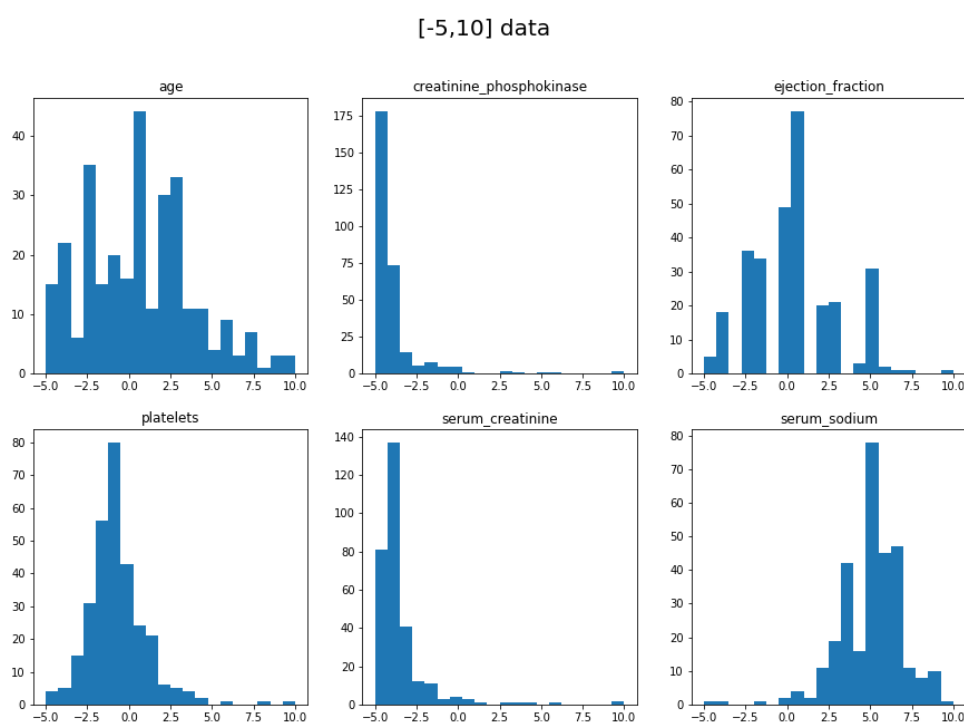


Рисунок 7 — Гистограмма после `range_5_10`

## Нелинейные преобразования

1. С помощью *QuantileTransformer* данные приведены к равномерному и нормальному распределениям (рис. 8-9).



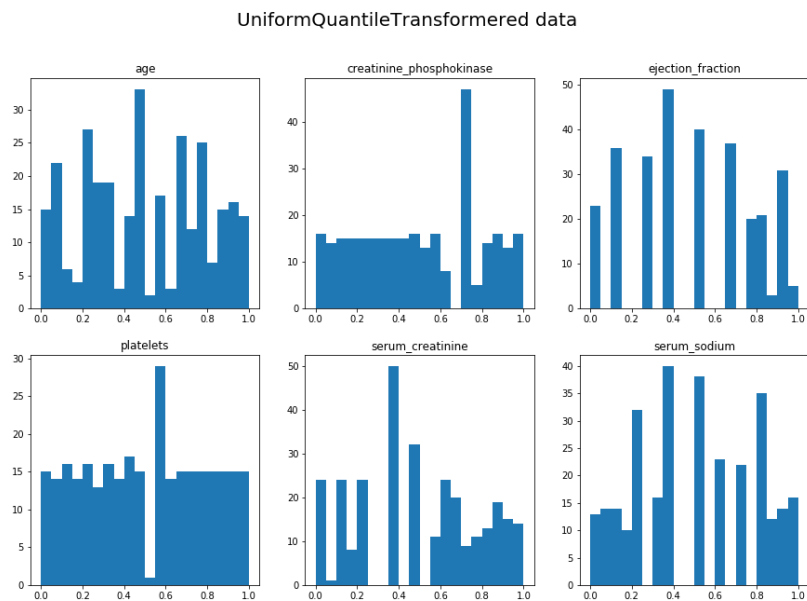


Рисунок 8 — Гистограмма после QuantileTransformer, равномерное распределение

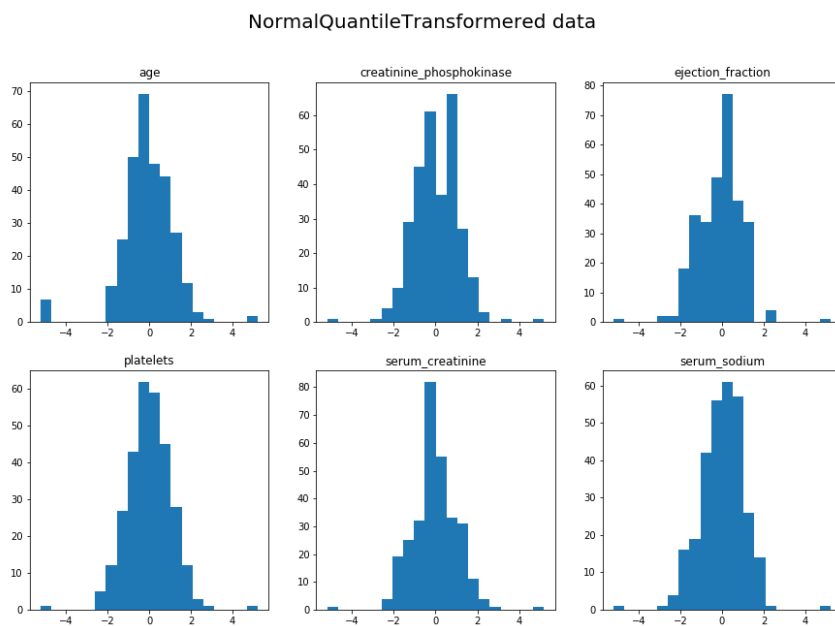


Рисунок 9 — Гистограмма после QuantileTransformer, нормальное распределение

$n\_quantiles$  — параметр, указывающий количество квантилей, используемых для дискретизации функции распределения, чем больше

количество квантилей, тем ближе полученная гистограмма к требуемому распределению.

2. С помощью *PowerTransformer* данные были приведены к нормальному распределению (рис. 10).



Рисунок 10 — Гистограмма после PowerTransformer

## Дискретизация признаков

1. Выполнена дискретизация признаков (рис. 11)

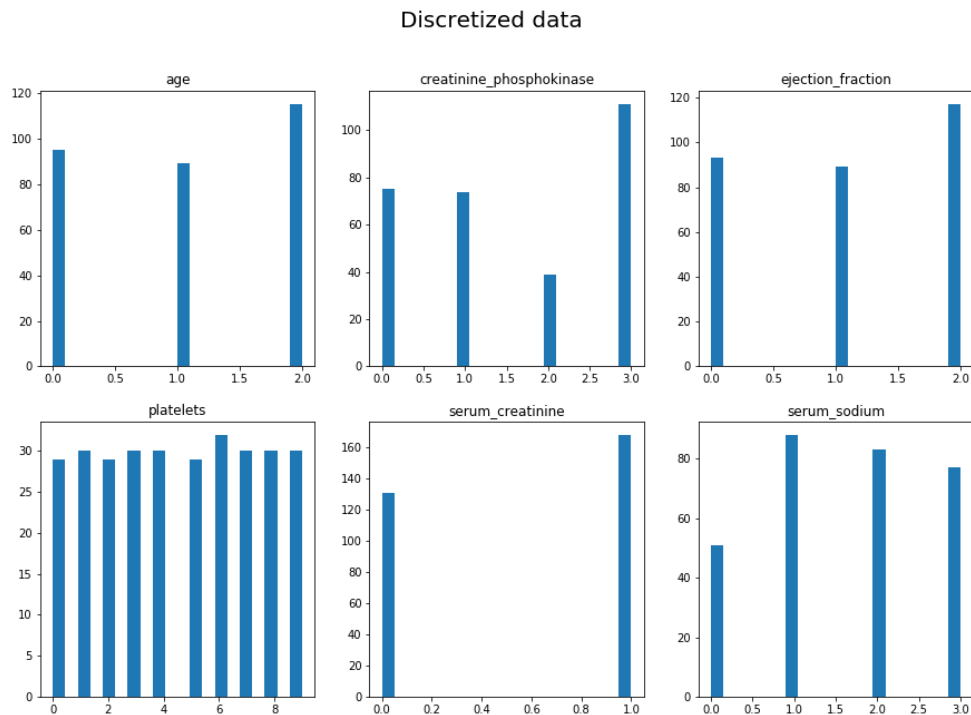


Рисунок 10 — Гистограмма после PowerTransformer

#### Диапазоны интервалов

- age: [40., 55., 65., 95.]
- creatinine\_phosphokinase: [ 23. , 116.5, 250. , 582. , 7861. ]
- ejection\_fraction: [14., 35., 40., 80.]
- platelets: [ 25100., 153000., 196000., 221000., 237000., 262000., 265000., 285200., 319800., 374600., 850000.]
- serum\_creatinine: [0.5, 1.1, 9.4]
- serum\_sodium: [113., 134., 137., 140., 148.]

#### Выводы

В ходе выполнения лабораторной работы изучены методы предобработки данных с помощью методов библиотеки Scikit Learn. При изучении стандартизации данных было выяснено, что при настройке на неполных данных происходит снижение качества результирующего набора данных, а приведение к диапазону не изменяет форму распределения. С

помощью нелинейных преобразований преобразована изначальная форма распределения.