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. 6307 _____ Трофимов Н.И.

Преподаватель _____ Жангиров Т.Р

Санкт-Петербург

2020

Цель работы.

Ознакомиться с методами предобработки данных из библиотеки Scikit Learn.

Ход работы

Загрузка данных

Датасет загружен в датафрейм pandas, удалены колонки: ['anaemia','diabetes','high_blood_pressure','sex','smoking','time','DEATH_EVENT'])

Построены гистограммы признаков, приведенные на рис. 1.

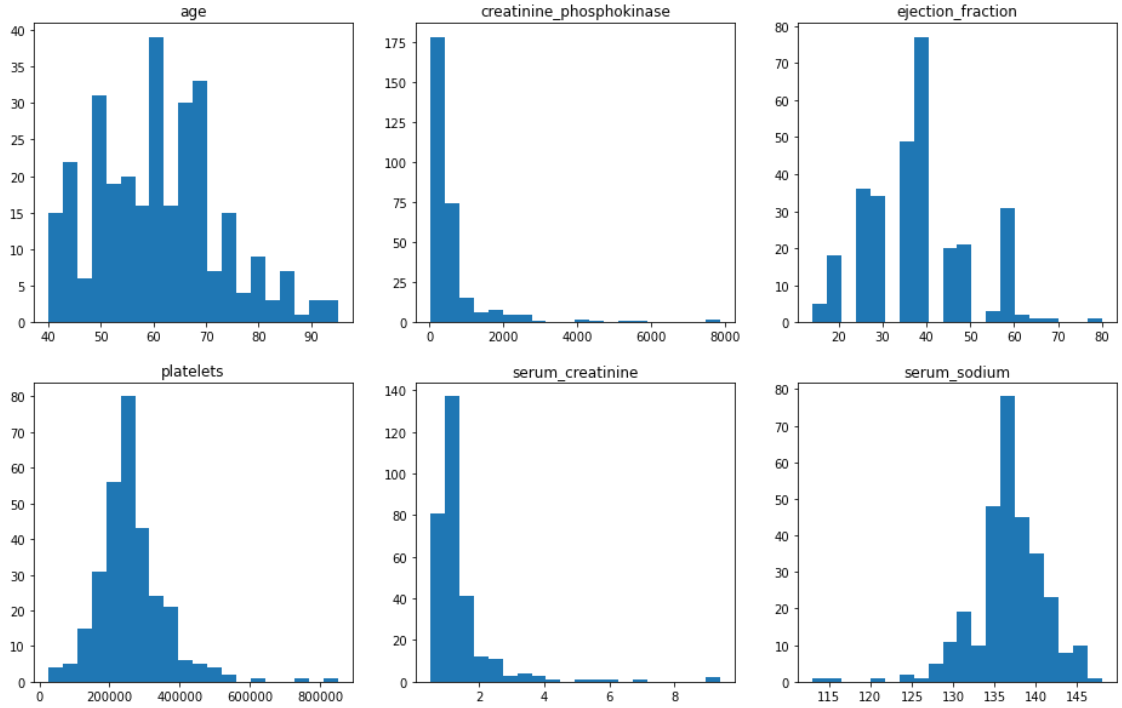

Рисунок 1. Гистограммы исходных данных

На основании гистограмм были определены диапазоны значений для каждого признака, а также найдена медиана. Данные приведены в таблице 1.

Признак	**Диапазон**	**Мода**
Age	40-95	60
creatinine_phosphokinase	23-7861	582
ejection_fraction	14-80	35
platelets	25100-850000	263358.03

serum_creatinine	0.5-9.4	1
serum_sodium	113-148	136

Стандартизация данных

Данные были стандартизованы с помощью StandardScaler на полном датасете и его срезе из 150 строк. На рис.2 изображены гистограммы стандартизированного датасета.

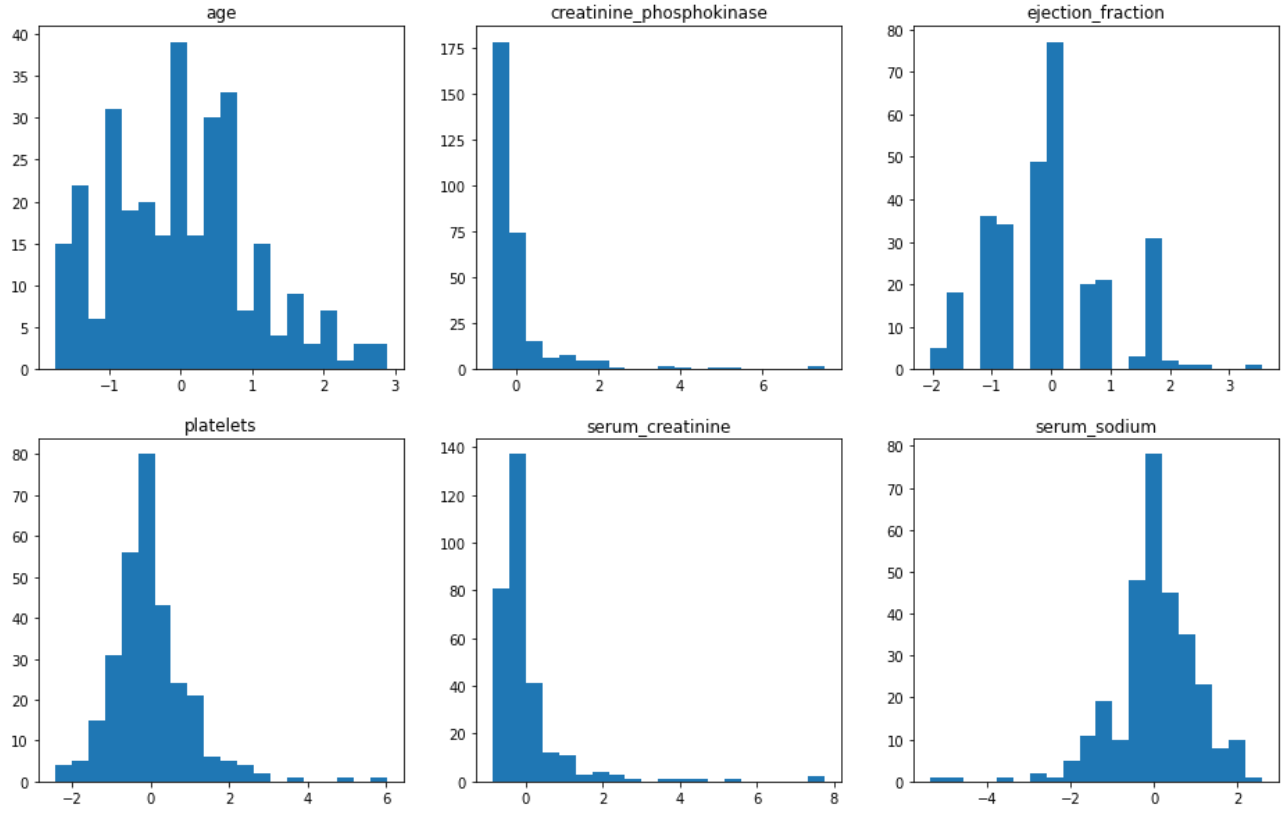

Рисунок 2. Стандартизированные данные

Также были рассчитаны мат ожидание и СКО до и после стандартизации. Результаты приведены ниже.

	age	creatinine_phosphokinase	ejection_fraction	platelets
mean	60.833893	581.839465	38.083612	263358.029264
std	11.894809	970.287881	11.834841	97804.236869

	serum_creatinine	serum_sodium
mean	1.39388	136.625418
std	1.03451	4.412477

Рисунок 3. Мат ожидание и СКО начальных данных

Мат ожидание и СКО при стандартизации на 150 данных:

```
Мат ожидание = [-0.16970362 -0.02127675 0.01050249 -0.03522879 -0.1086408
0.0379076 ]
СКО = [0.95382379 0.81417905 0.90610822 1.01506113 0.88542887 0.9703736 ]
```

Мат ожидание и СКО для стандартизации на полном датасете:

```
Мат ожидание = [ 5.70335306e-16  0.00000000e+00 -3.26754603e-17
7.72329061e-17  1.42583827e-16 -8.67384945e-16]
СКО = [1. 1. 1. 1. 1. 1.]
```

Формула, по которой стандартизировались признаки: Z = (Xi - M)/Std, где M - мат ожидание, std – СКО.

Сравнение значений из формул с полями mean_ и var_ объекта scaler:

Признак	Var_	Mean_
age	141	60.8
creatinine_phosphokinase	938309.8	581.8
ejection_fraction	139.5	38.1
platelets	953367655	263358
serum_creatinine	1	1.4
serum_sodium	136.4	136.6

Приведение к диапазону

Чтобы привести данные к диапазону [0, 1], использовался MinMaxScale, гистограммы изображены на рис. 3.

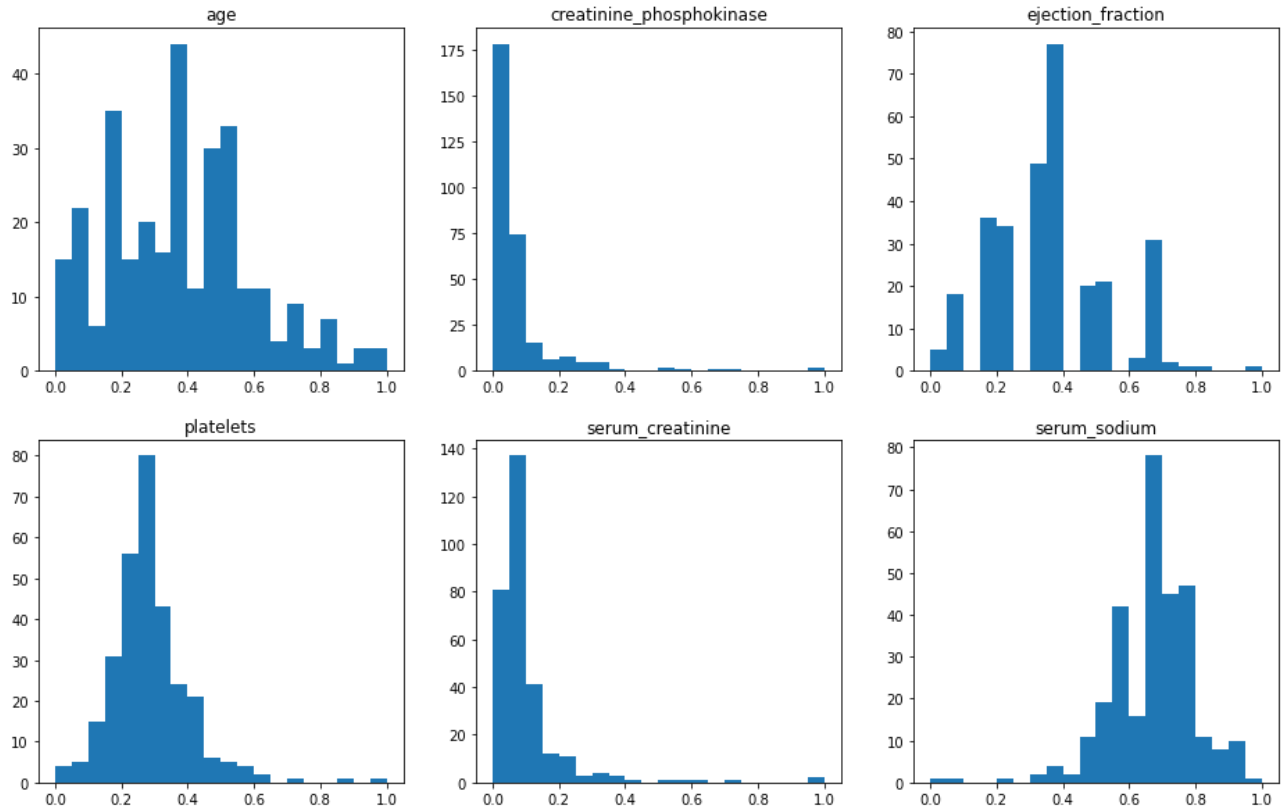

Рисунок 4. Гистограммы в новом диапазоне

Минимальные и максимальные значения для каждого признака представлены

ниже.

```
Min value for each column - [4.00e+01 2.30e+01 1.40e+01 2.51e+04 5.00e-01 1.13e+02]
Max value for each column - [9.500e+01 7.861e+03 8.000e+01 8.500e+05 9.400e+00 1.480e+02]
```

Данные, преобразованные с помощью MaxAbsScaler и RobustScaler представлены на рисунках 5 и 6.

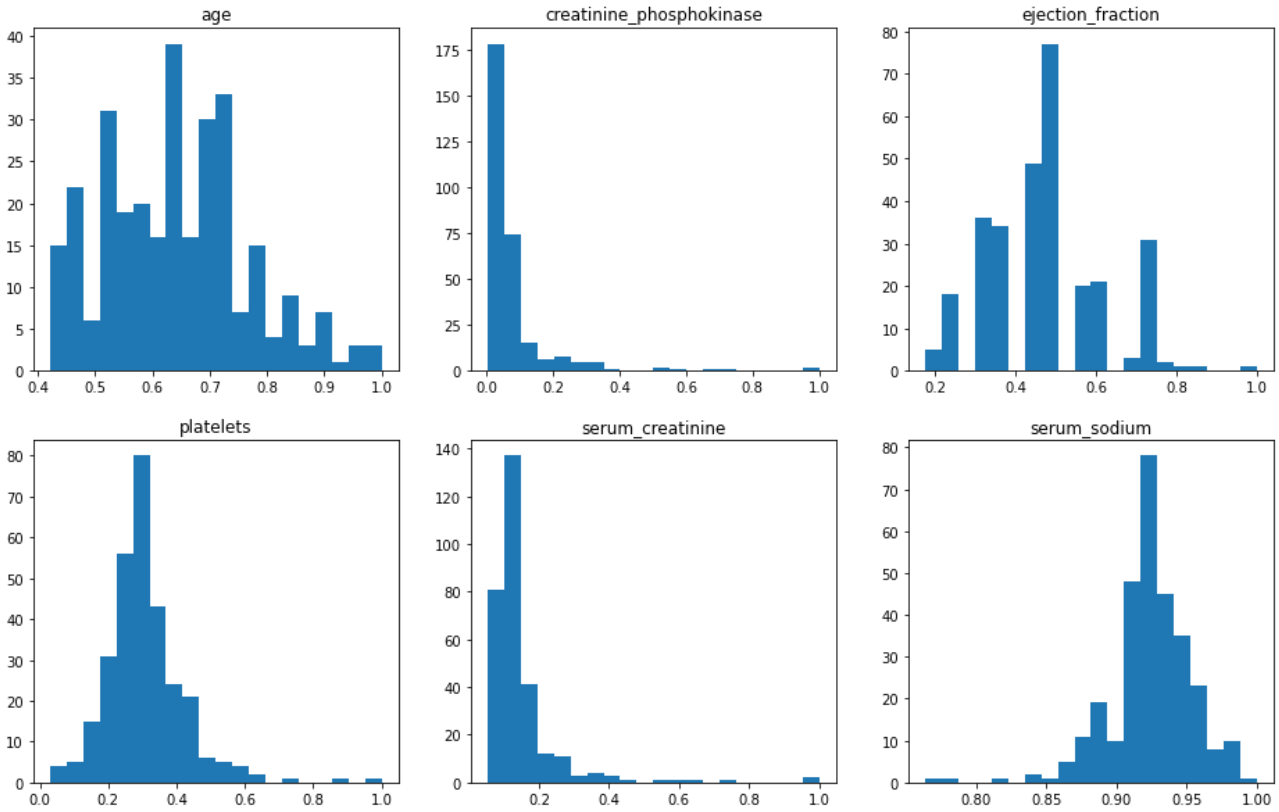

Рисунок 5 MaxAbsScaler

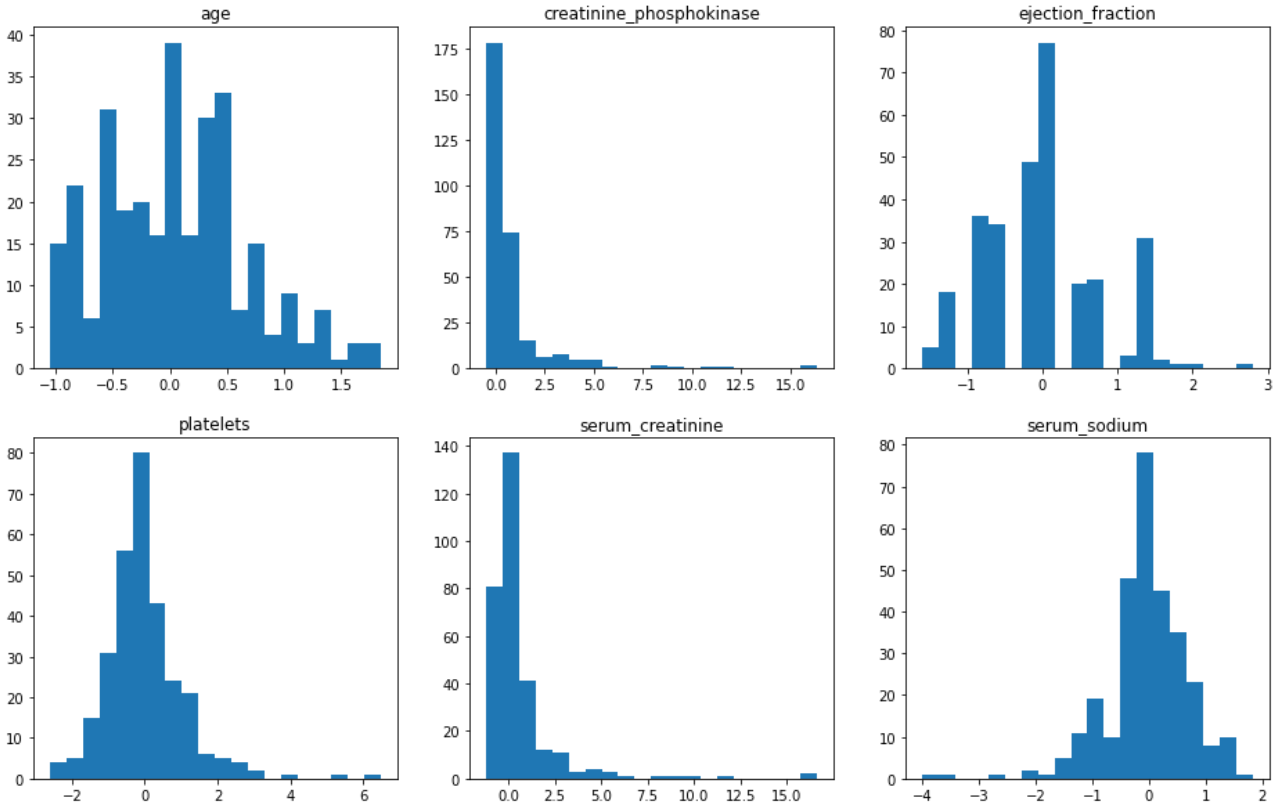

Рисунок 6 RobustScaler

Функция, приводящая все данные к диапазону [-5, 10]:

```
def set_range(data):
    scaler = preprocessing.MinMaxScaler(feature_range=[-5, 10]).fit(data)
    return scaler.transform(data)
```

Нелинейные преобразования

С помощью QuantileTransformer данные были приведены к равномерному распределению. Построенные гистограммы изображены на рисунке 7.

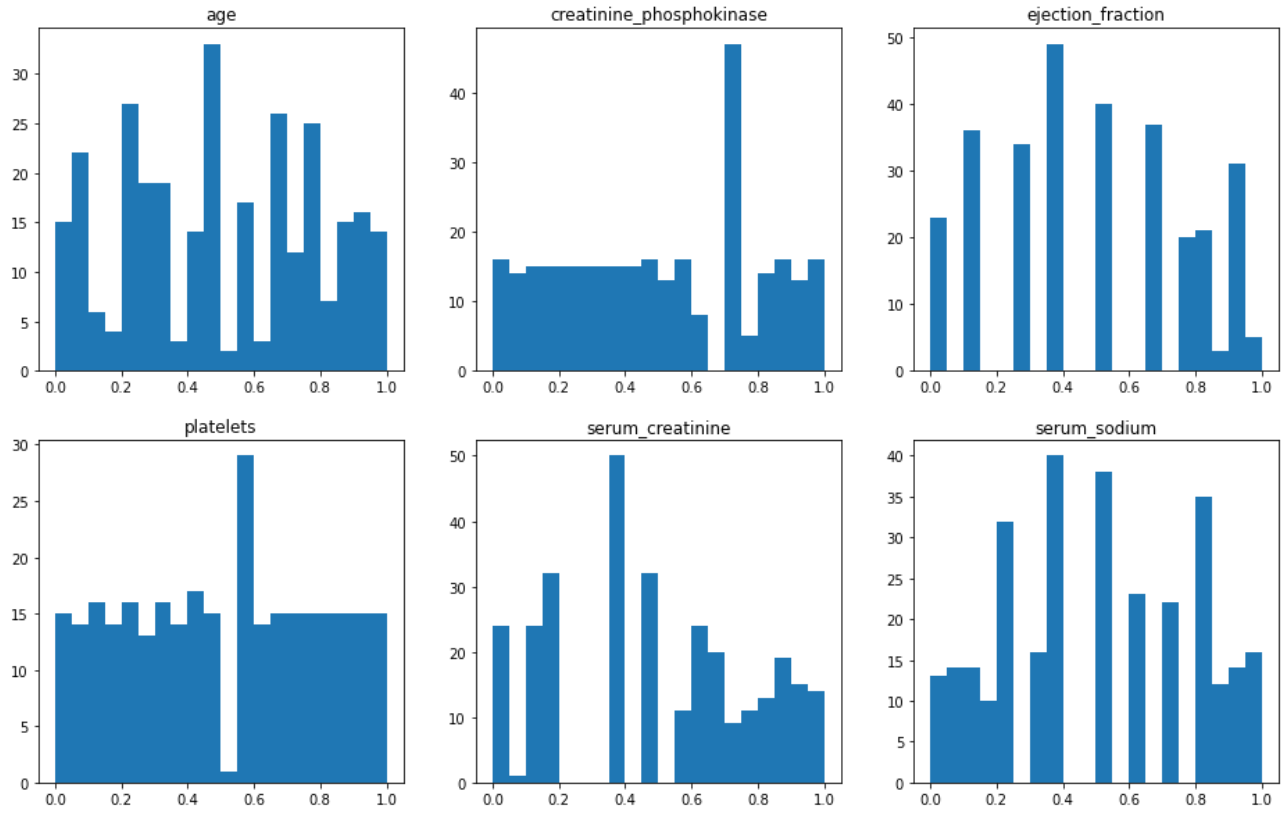

Рисунок 7 Равномерное распределение

Параметр n_quantilies определяет количество вычисляемых квантилей. Оно не может быть больше, чем число наблюдений.

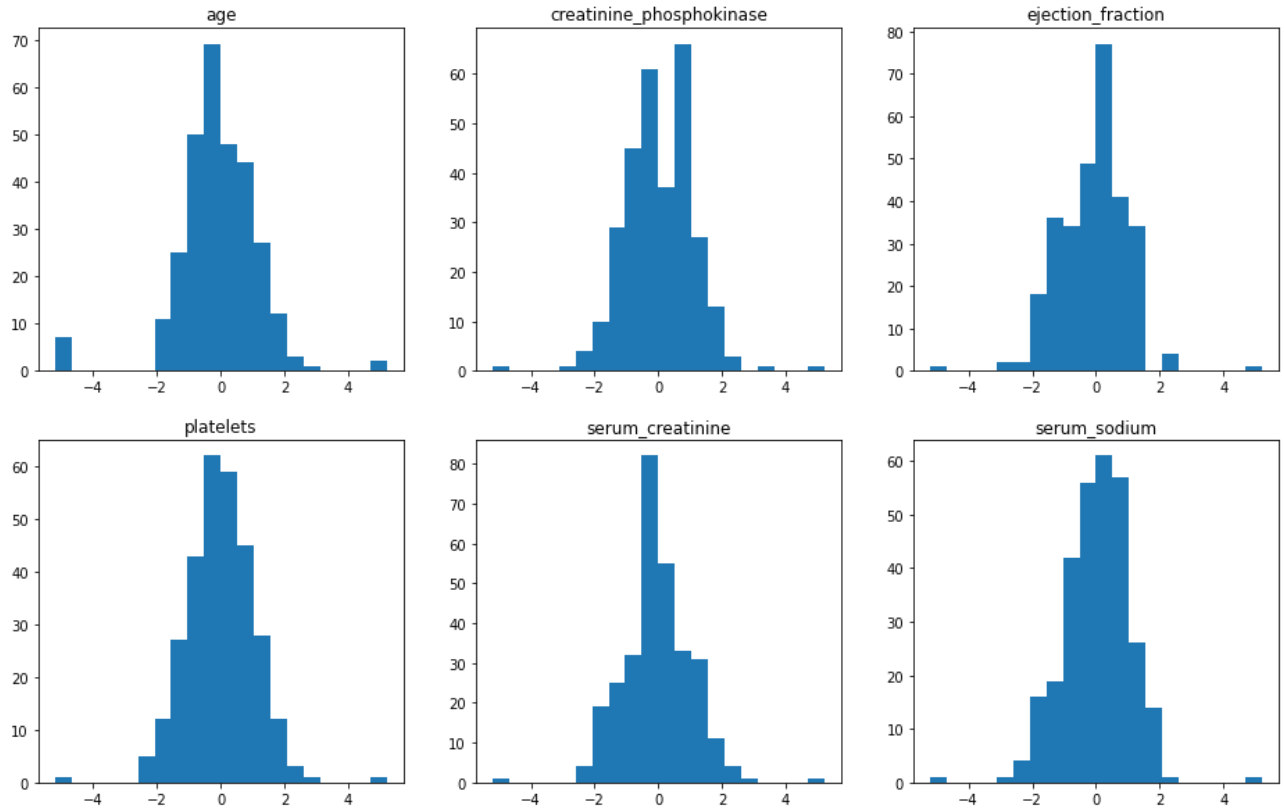

Рисунок 8 Нормальное распределение через QuantileTransformer

Также данные были приведены к нормальному распределению через PowerTrnsformer.

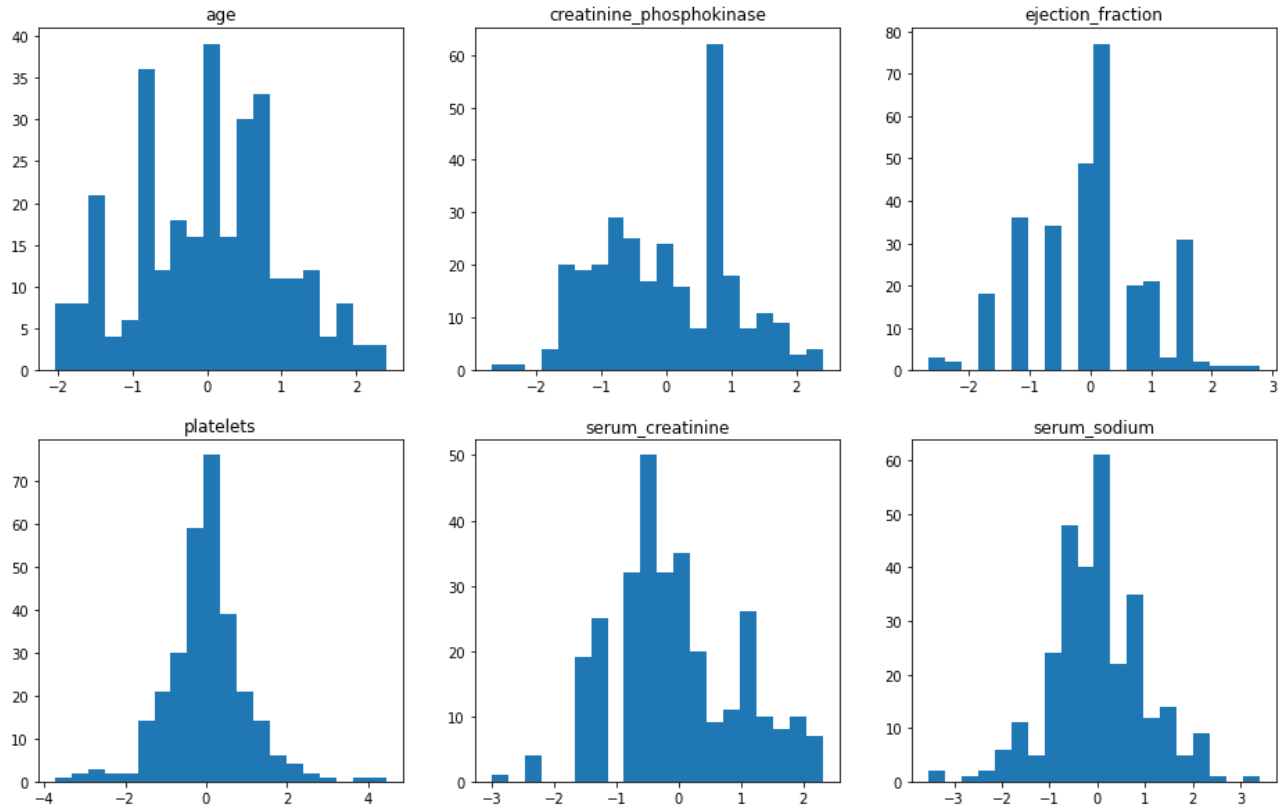

Рисунок 9 PowerTransformer

Дискретизация признаков

Дискретизированные данные с заданным количеством диапазонов через KBinsDiscretizer представлены на рисунке 10.

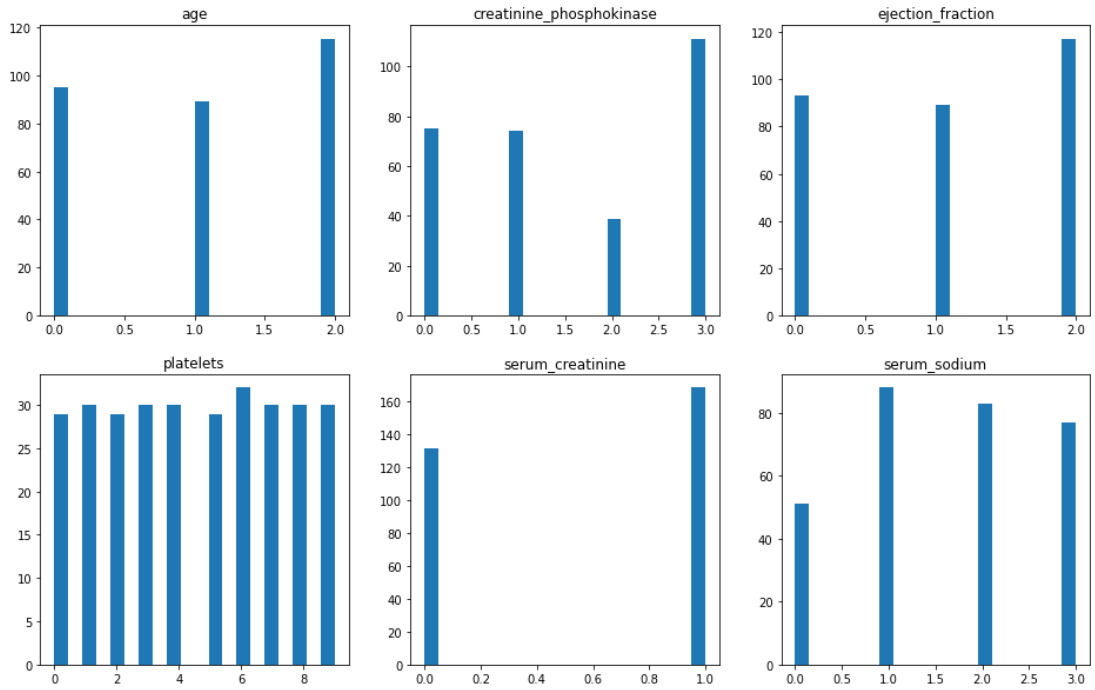

Рисунок 10 Дискретизированные данные

Значения полученных диапазонов хранятся в bin_edges_:

```
[array([40., 55., 65., 95.]),
 array([  23. ,  116.5,  250. ,  582. , 7861. ]),
 array([14., 35., 40., 80.]),
 array([ 25100., 153000., 196000., 221000., 237000., 262000., 265000.,
 285200., 319800., 374600., 850000.]),
 array([0.5, 1.1, 9.4]), array([113., 134., 137., 140., 148.])],
```

Выводы

В ходе работы были получены навыки по предобработке данных методами библиотеки Scikit Learn, позволяющих выполнить стандартизацию, привидение к диапазонам, нелинейные преобразования и дискретизацию данных.