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 _____ Медведев Е. Р.

Преподаватель _____ Жангиров Т. Р.

Санкт-Петербург

2020

Цель работы

Ознакомиться с методами предобработки данных из библиотеки Scikit-Learn.

Ход работы

Данные загружены из csv файла, столбцы, содержащие бинарные признаки и признаки времени, исключены.

	age	creatinine_phosphokinase	...	serum_creatinine	serum_sodium
0	75.0	582	...	1.9	130
1	55.0	7861	...	1.1	136
2	65.0	146	...	1.3	129
3	50.0	111	...	1.9	137
4	65.0	160	...	2.7	116
..	...	...	...	...	...
294	62.0	61	...	1.1	143
295	55.0	1820	...	1.2	139
296	45.0	2060	...	0.8	138
297	45.0	2413	...	1.4	140
298	50.0	196	...	1.6	136

[299 rows x 6 columns]

Рис. 1 – Загруженные данные

Построены гистограммы признаков:

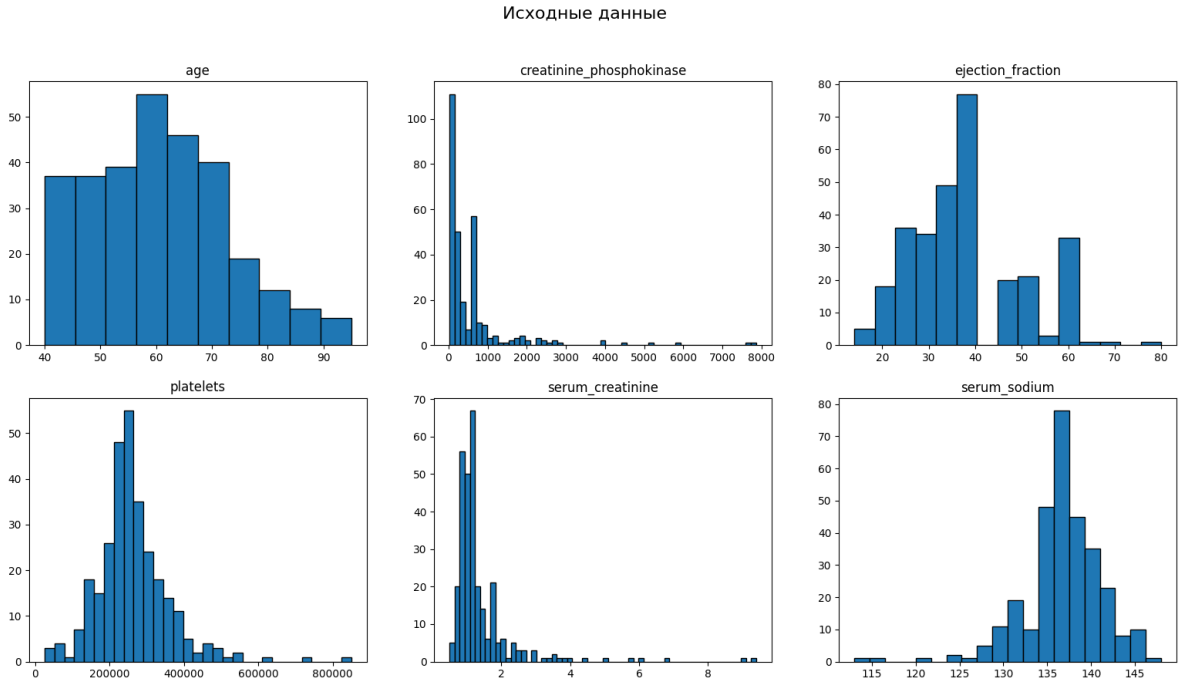

Рис. 2 – Гистограммы исходных данных

Из гистограмм видны диапазоны значений признаков, а также где лежит наибольшее количество наблюдений:

Признак	Диапазон значений	Скопление наблюдений
age	[40; 95]	60
creatinine_phosphokinase	[0; 7800]	100
ejection_fraction	[15; 80]	40
platelets	[25000; 850000]	260000
serum_creatinine	[0.5; 9.5]	1
serum_sodium	[113; 148]	136

Стандартизация данных

С помощью StandardScaler настроена стандартизация данных на основе первых 150 наблюдений.

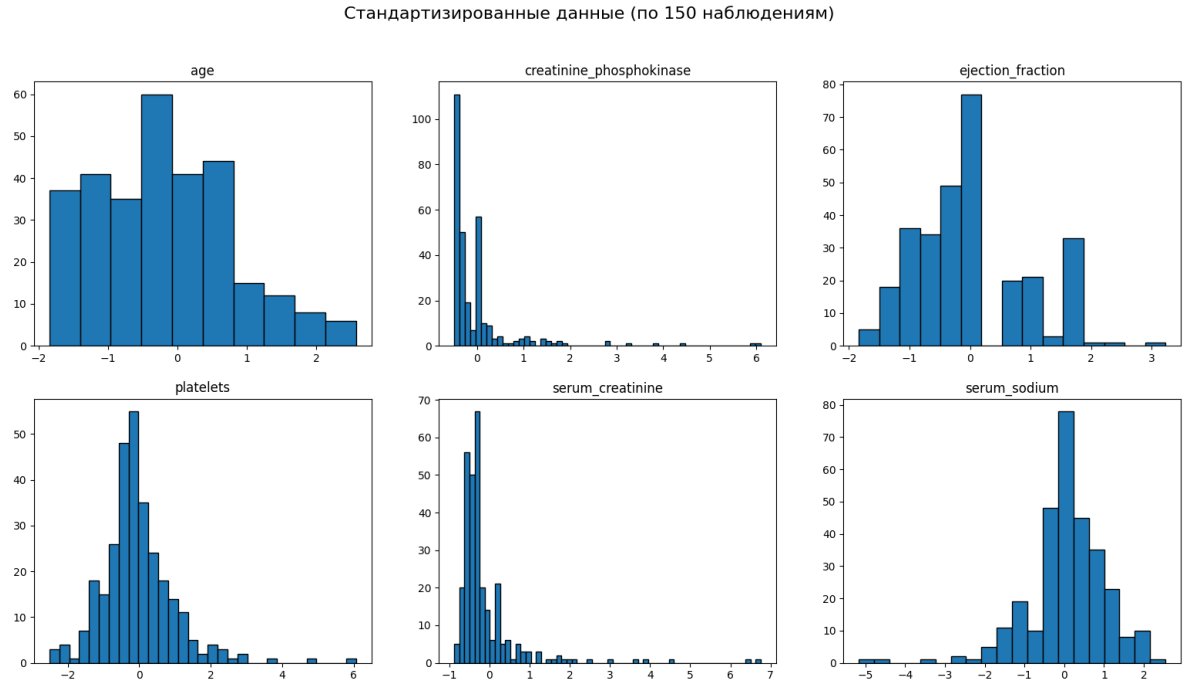

Рис. 3 – Стандартизация StandardScaler (150 наблюдений)

Сравнивая полученные гистограммы с исходными, можно увидеть, что диапазоны значений изменились, а значение, в котором наблюдений большинство, на всех гистограммах стремится к нулю.

При стандартизации на основе всех данных этот эффект далее усиливается.

Формула для стандартизации, используемая StandardScaler:

$$X_{scaled} = (X - M[X])/\text{СКО}[X]$$

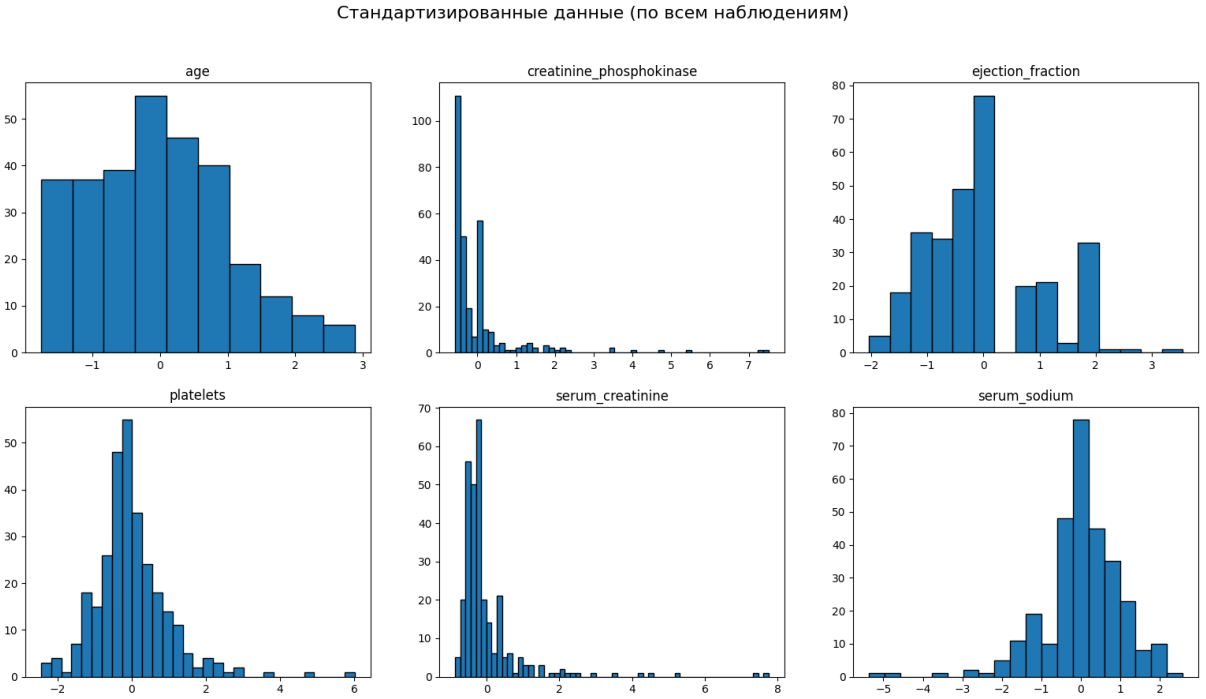

Рис. 4 – Стандартизация StandardScaler (по всем наблюдениям)

Таблица 1 – сводная таблица мат. ожиданий и СКО в разных тестах

	mean non scaled	mean 150	mean 150 scalar	mean	mean scalar	std non scaled	std 150	std 150 scalar	std	std scalar
age	60,834	-0,170	62,947	0,000	60,834	11,875	0,954	12,450	1,000	11,875
creatinine_phosphokinase	581,839	-0,021	607,153	0,000	581,839	968,664	0,814	1189,743	1,000	968,664
ejection_fraction	38,084	0,011	37,947	0,000	38,084	11,815	0,906	13,039	1,000	11,815
platelets	263358,029	-0,035	266746,749	0,000	263358,029	97640,548	1,015	96191,790	1,000	97640,548
serum_creatinine	1,394	-0,109	1,521	0,000	1,394	1,033	0,885	1,166	1,000	1,033
serum_sodium	136,625	0,038	136,453	0,000	136,625	4,405	0,970	4,540	1,000	4,405

Приведение к диапазону

С помощью MinMaxScaler данные приведены к диапазону [0; 1].

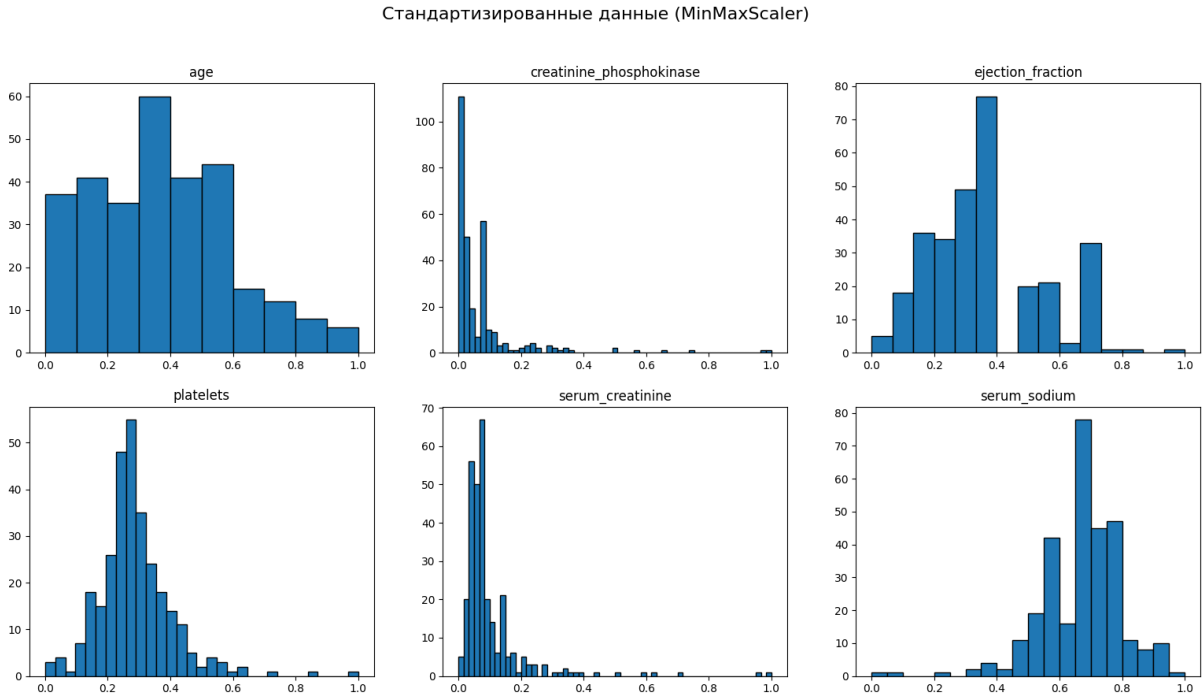

Рис. 5 – Приведение к диапазону MinMaxScaler

Для случая min=0, max=1 формула будет выглядеть так:

$$X_{scaled} = (X - min(X)) / (max(X) - min(X))$$

По полям объекта MinMaxScaler определили максимальные и минимальные значения для признаков.

Минимум: 40.0 23.0 14.0 25100.0 0.5 113.0

Максимум: 95.0 7861.0 80.0 850000.0 9.4 148.0

MaxAbsScaler работает похоже (рисунок 6), приводя данные к диапазону [-1; 1] на основании максимального модуля по формуле:

$$X_{scaled} = X / max(abs(X))$$

Максимальный модуль из поля объекта MaxAbsScaler:

95.0 7861.0 80.0 850000.0 9.4 148.0

Еще один похожий объект RobustScaler стандартизирует данные по межквартильному размаху (межквартильный размах – это разница между 3-м и 1-м квартилями. У данного показателя есть одно неоспоримое преимущество: он является робастным, т. е. не зависит от аномальных отклонений). Формула преобразования:

$$X_{scaled} = (X - Q_1(X)) / (Q_3(X) - Q_1(X))$$

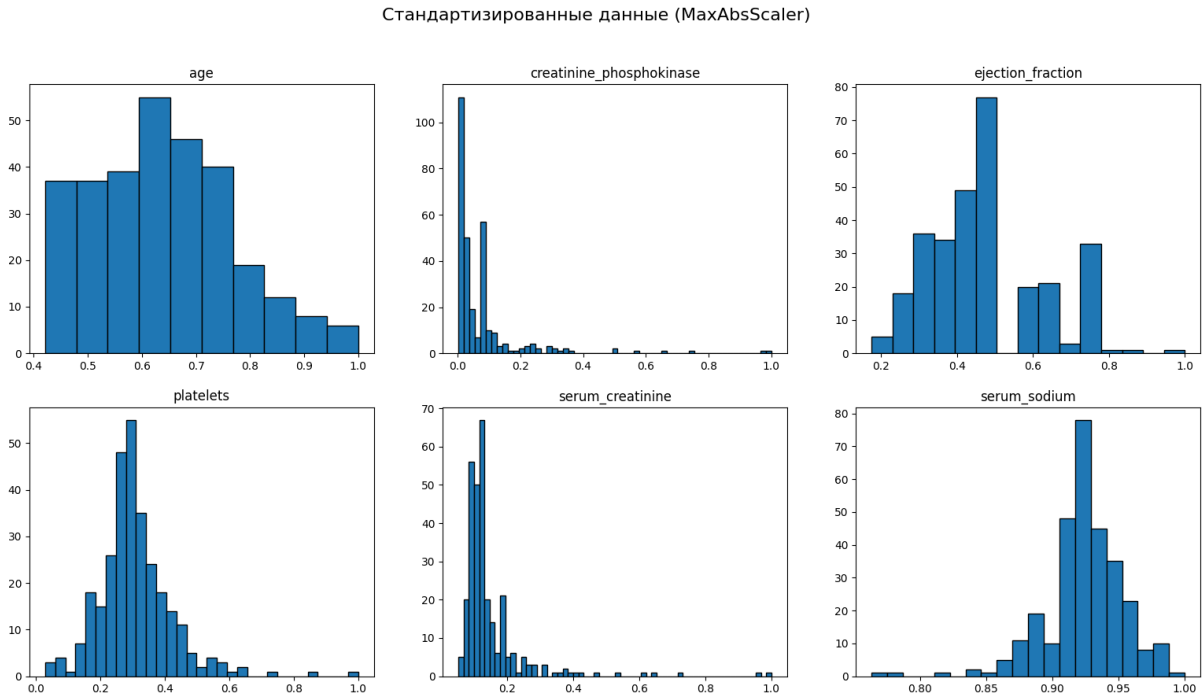

Рис. 6 – Приведение к диапазону MaxAbsScaler

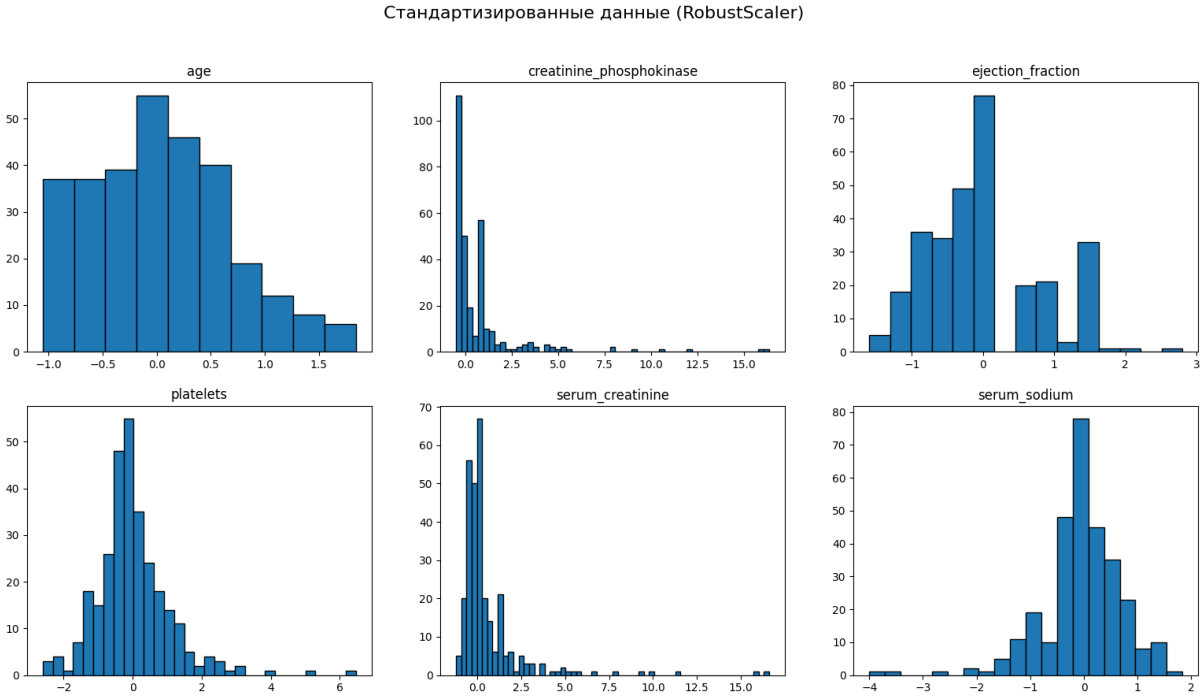

Рис. 7 – Приведение к диапазону RobustScaler

На основе объекта MinMaxScaler была создана функция, приводящая данные к диапазону [-5; 10] по формуле

$$X_{scaled} = (X - min(X)) \,/\, (max(X) - min(X)) * (max - min) + min,$$

где max=10, min=5.

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

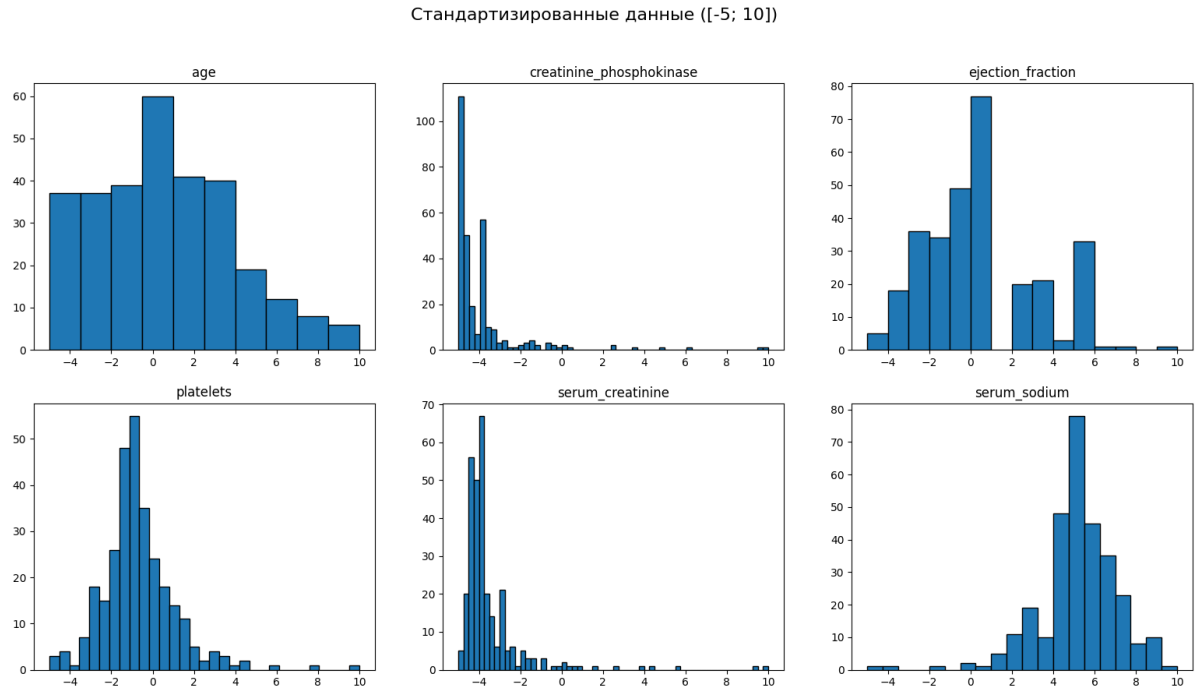

Рис. 8 – Приведение к диапазону [-5; 10]

Нелинейные преобразования

Данные приведены к равномерному распределению по 100 квантилям:

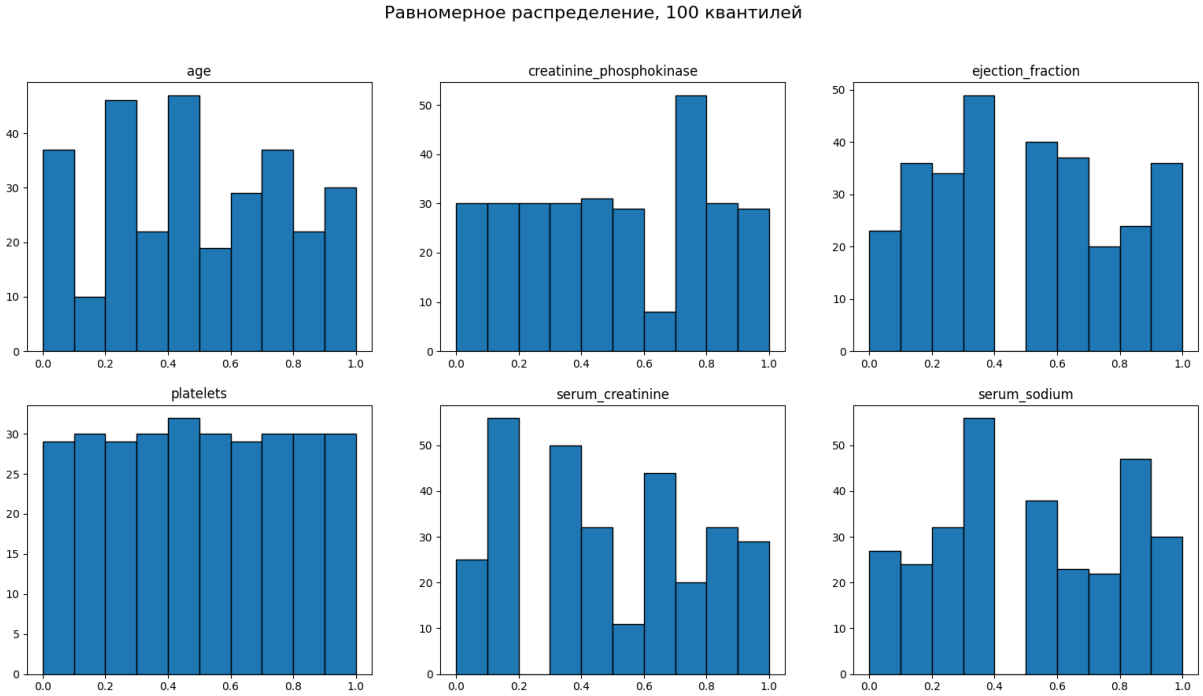

Рис. 9 – Равномерное распределение (100 квантилей)

Видно, что данные стали стремиться к равномерному распределению. Количество квантилей, используемых функцией, задает точность дискретизации функции распределения, что влияет на то, как хорошо данные

будут приближены к ней. С меньшим числом квантилей качество преобразования ухудшилось.

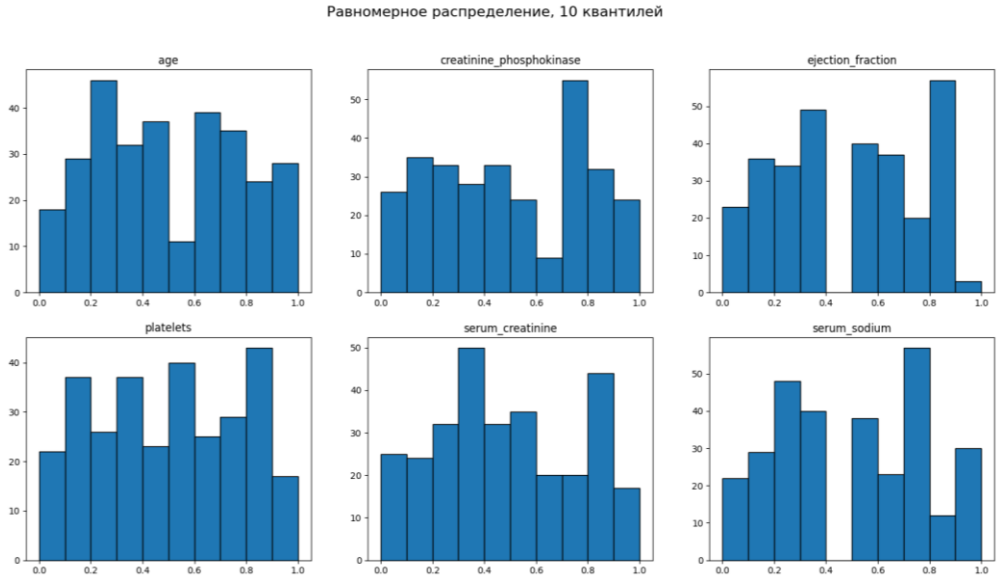

Рис. 10 – Равномерное распределение (10 квантилей)

Используя этот же объект можно привести данные к нормальному распределению:

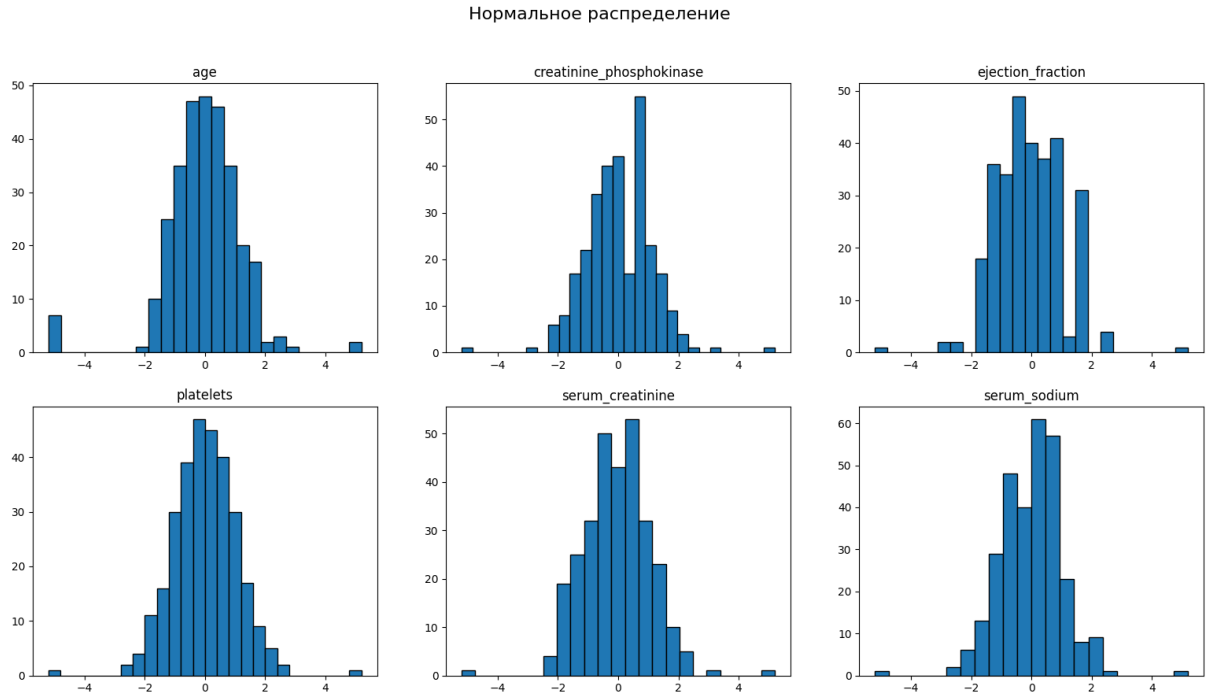

Рис. 11 – Нормальное распределение

Другой вариант приведения к нормальному распределению – с помощью объекта PowerTransformer.

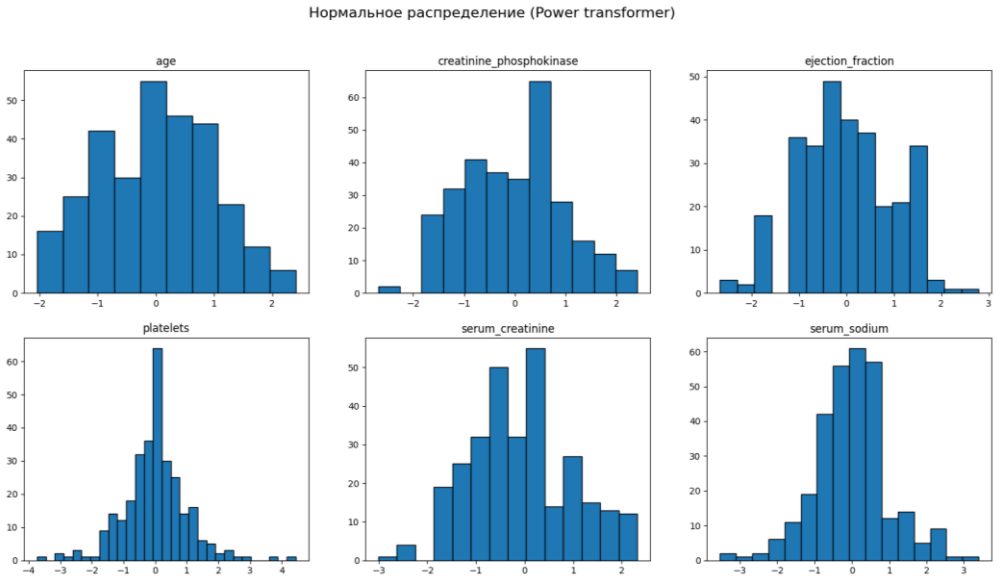

Рис. 12 – Нормальное распределение (PowerTransformer)

Дискретизация признаков

При помощи KBinsDiscretizer данные по всем признакам были дискретизированы на заданное количество диапазонов.

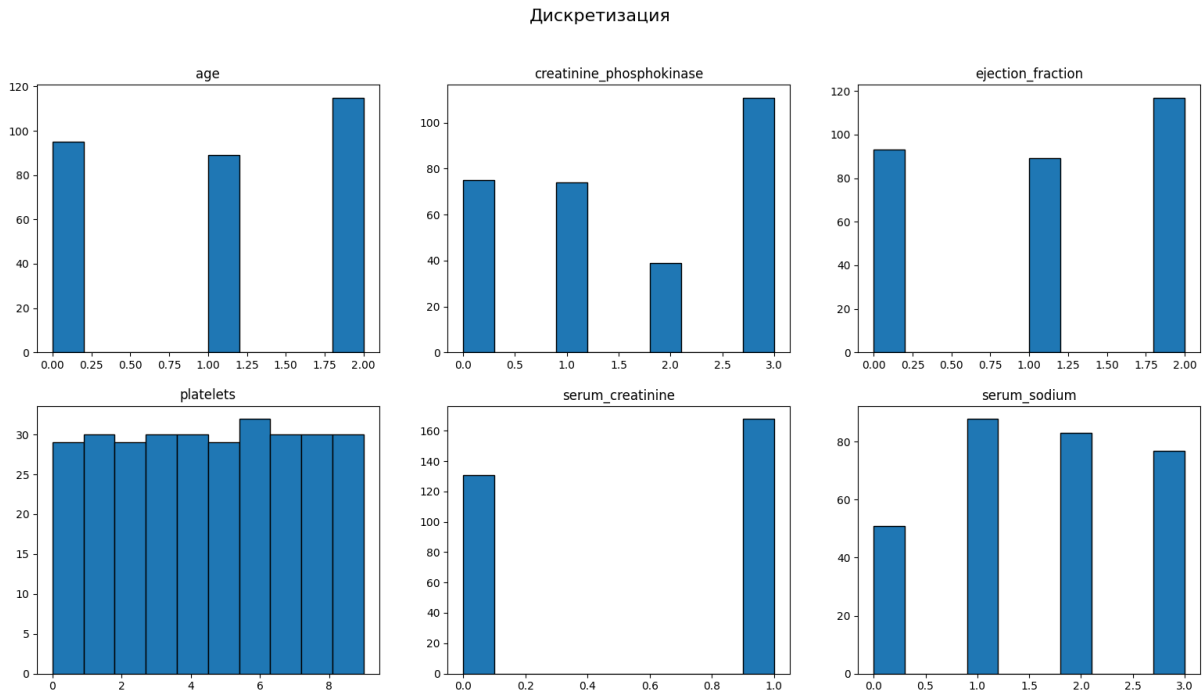

Рис. 13 – Дискретизация данных

Теперь диапазон значений на гистограммах представляет собой индексы дискретных значений. Заданный способ кодирования в данном случае – ordinal, поэтому диапазоны пронумерованы индексами.

Края диапазонов из поля объекта:

age: [40.0 55.0 65.0 95.0]

creatinine_phosphokinase: [23.0 116.5 250.0 582.0 7861.0]

ejection_fraction: [14.0 35.0 40.0 80.0]

platelets: [25100.0 153000.0 196000.0 221000.0 237000.0 262000.0 265000.0 285200.0 319800.0 374600.0 850000.0]

serum_creatinine: [0.5 1.1 9.4]

serum_sodium: [113.0 134.0 137.0 140.0 148.0]

Вывод

В результате выполнения лабораторной работы были изучены методы предобработки данных из библиотеки Scikit-learn.