**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. 8304 |  | Кирьянов Д. И. |
| Преподаватель |  | Жангиров Т.Р. |

Санкт-Петербург

2021

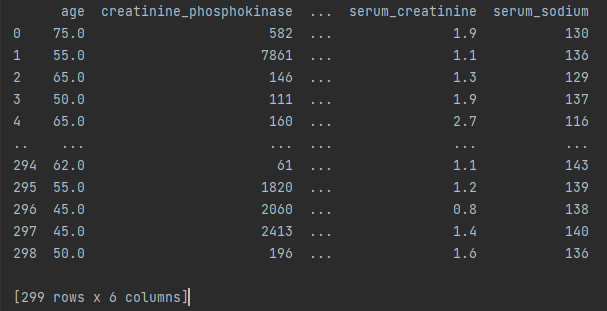
**Цель работы.**

Ознакомиться с методами предобработки данных из библиотеки Scikit

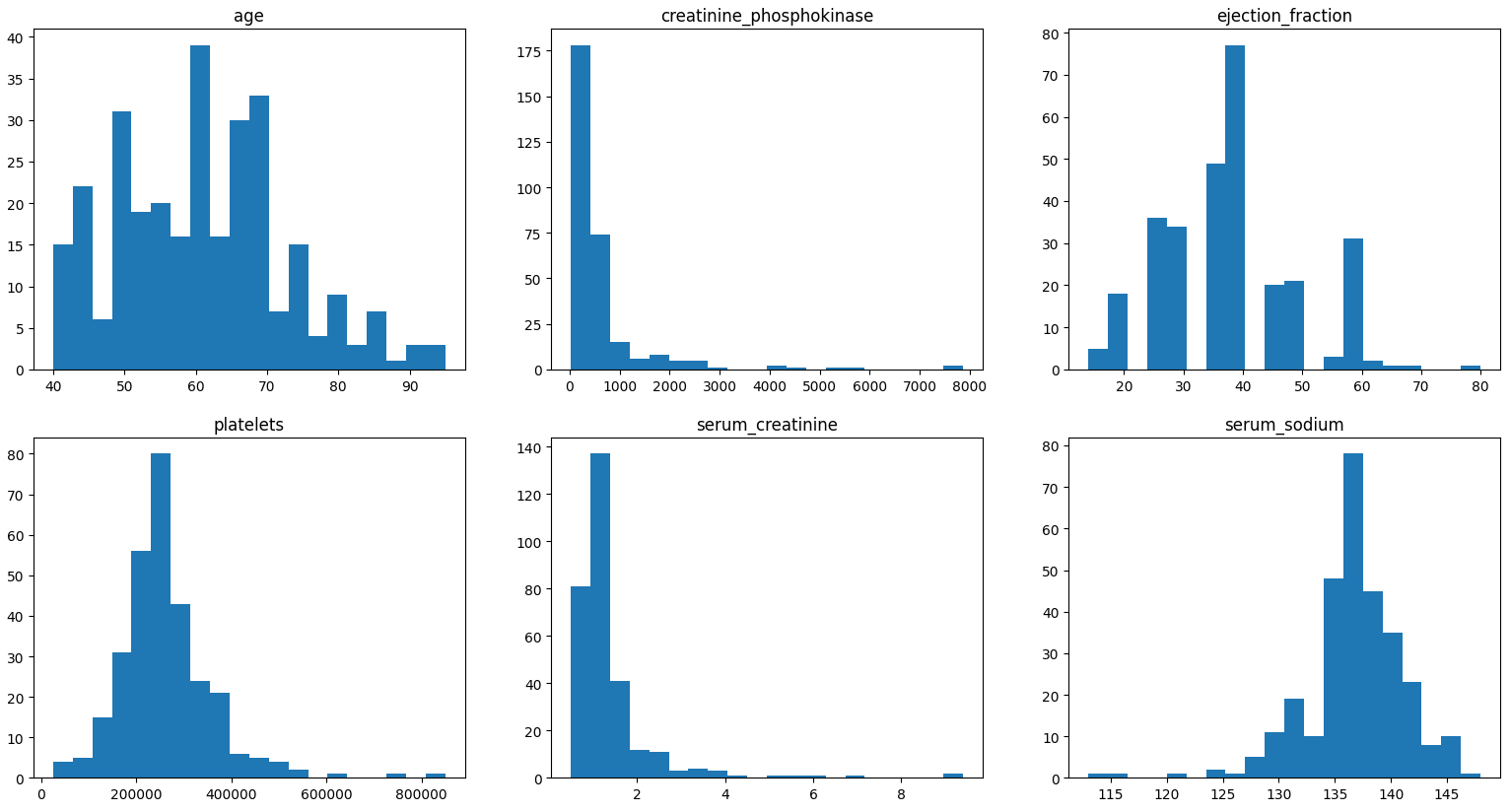
Learn.

**Ход работы.**

1. **Загрузка данных**
   1. Создан Python скрипт. Загружен датасет в датафрейм, и исключены бинарные признаки и признак времени.



* 1. Построены гистограммы признаков



age: диапазон [40; 95], наибольшее кол-во наблюдений: [59.25; 62]

creatinine\_phosphokinase: диапазон [23; 7861], наибольшее кол-во наблюдений: [23; 414.9]

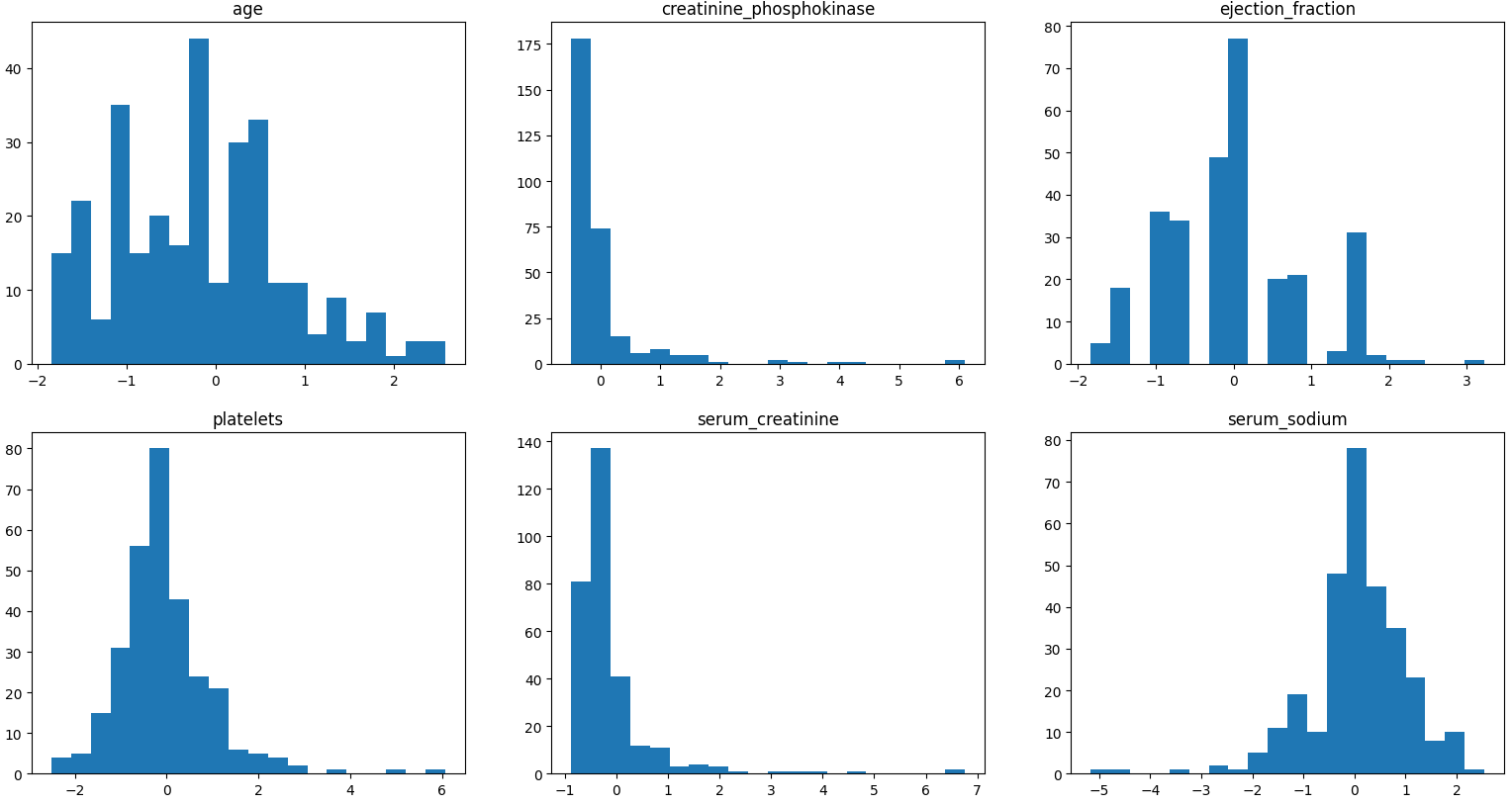
ejection\_fraction: диапазон [14; 80], наибольшее кол-во наблюдений: [37.1; 40.4]

platelets: диапазон [25000; 850000], наибольшее кол-во наблюдений: [231325; 272570]

serum\_creatinine: диапазон [0.5; 9.4], наибольшее кол-во наблюдений: [0.945; 1.39]

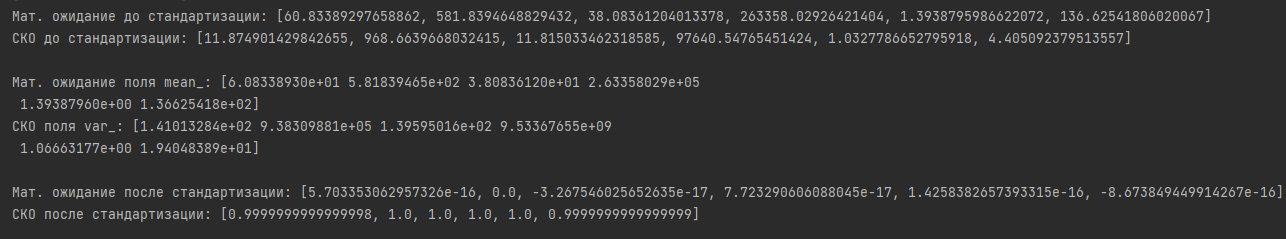
serum\_sodium: диапазон [113; 148], наибольшее кол-во наблюдений: [135.75; 137.5]

1. **Стандартизация данных**
   1. Подключен модуль Sklearn. Настроена стандартизация на основе первых 150 наблюдений используя StandardScaler и стандартизированы все данные. Построены гистограммы стандартизированных данных:



Уменьшился диапазон значений, т.к. стандартизация приводит исходные значения к среднему = 0 и стандартному отклонению = 1.

* 1. Рассчитано мат. ожидание и СКО до и после стандартизации, а также выведены значения полей mean\_ и var\_ объекта scaler.



Значения поля mean\_ совпадают со значениями из формул.

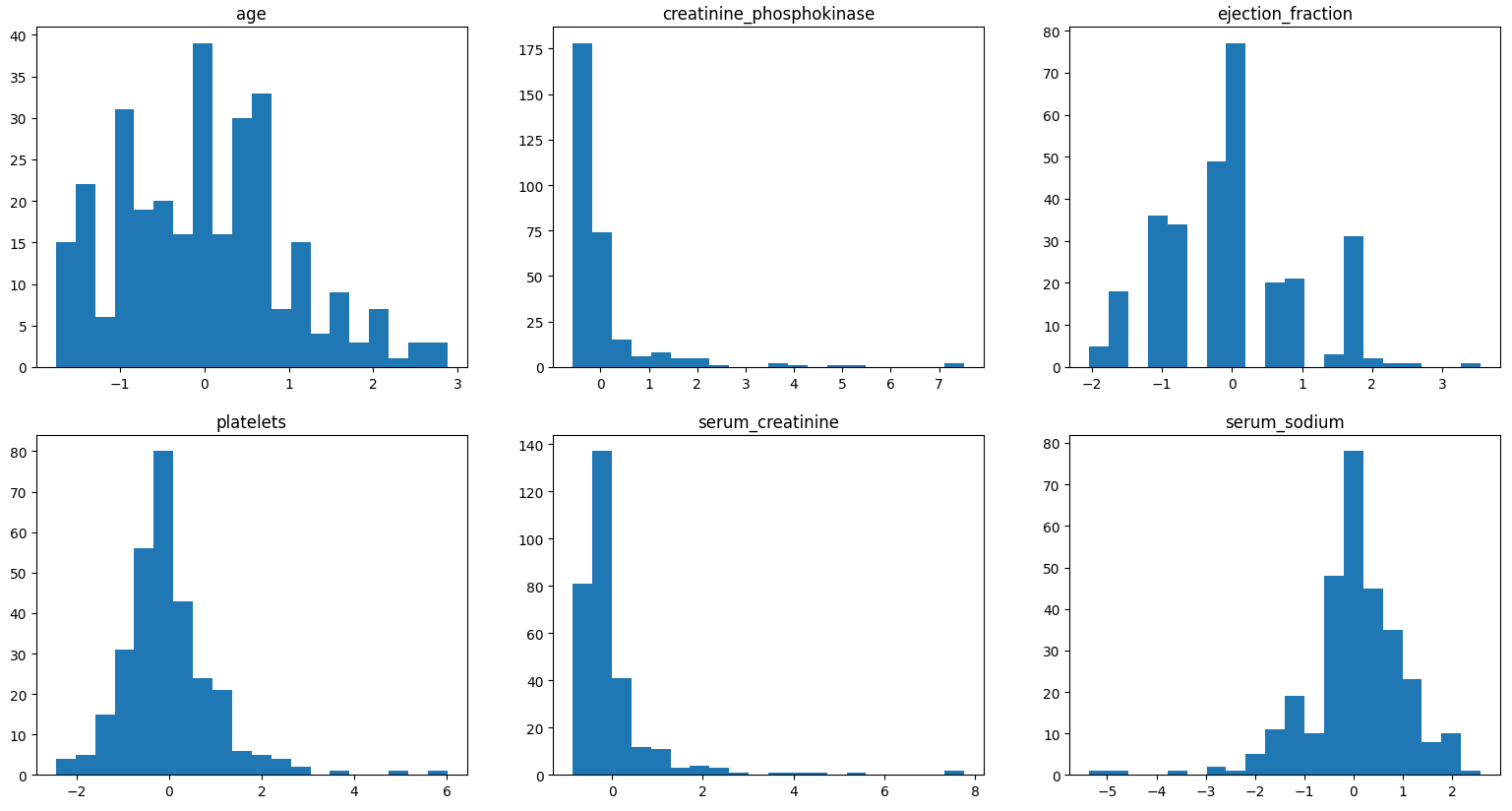
Значения поля var\_ не совпадают со значениями из формул.

Формула standardScaler:

z = (x - u) / s,

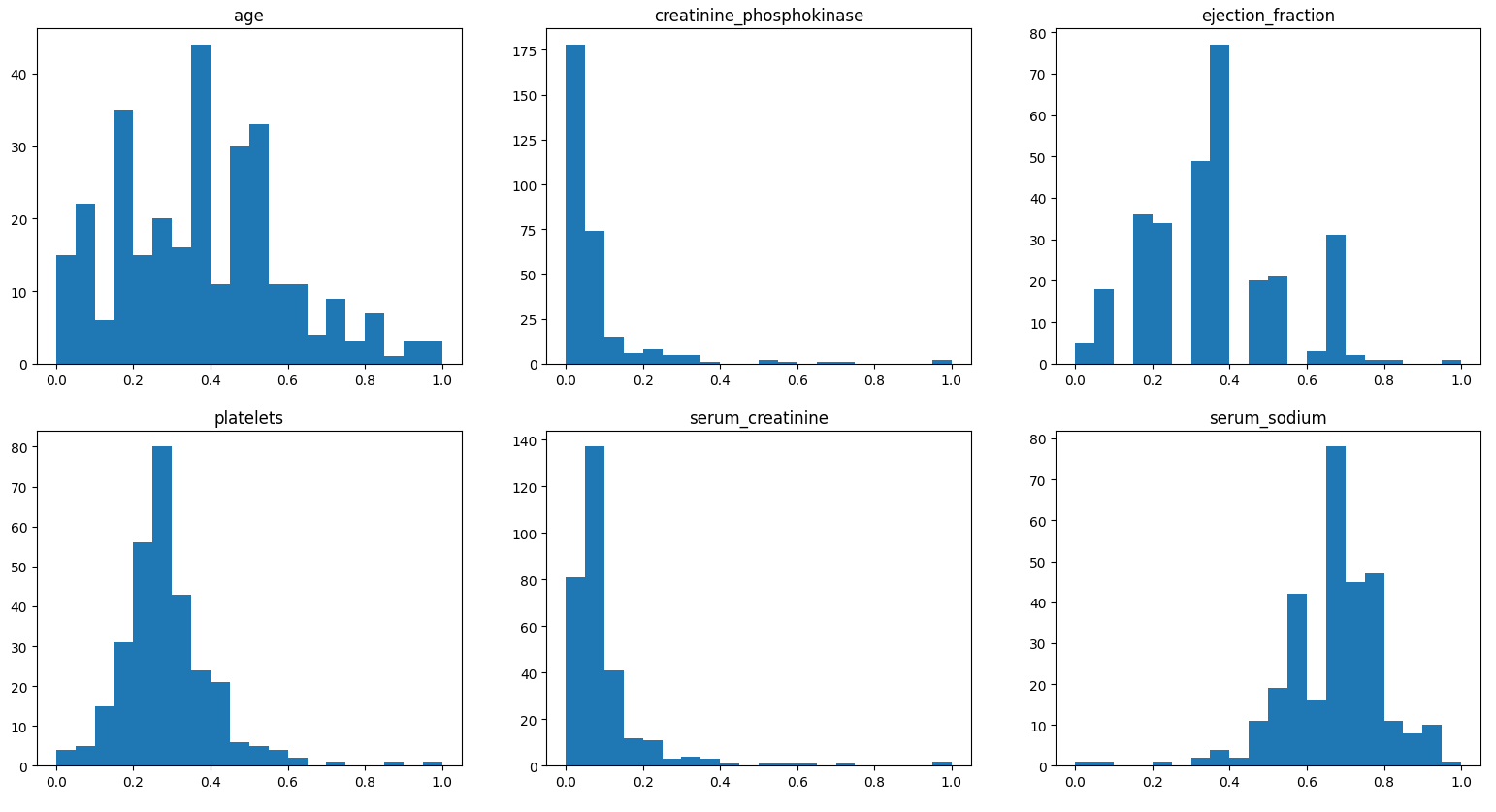
где u – мат. ожидание, s – СКО, x – обрабатываемое значение.

* 1. Проведена настройка стандартизации на всех данных.



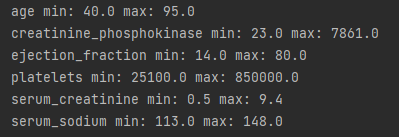
Диапазоны отличаются по сравнению со стандартизацией на основе первых 150 наблюдений.

1. **Приведение к диапазону**
   1. Данные приведены к диапазону с помощью MinMaxScaler, построены гистограммы



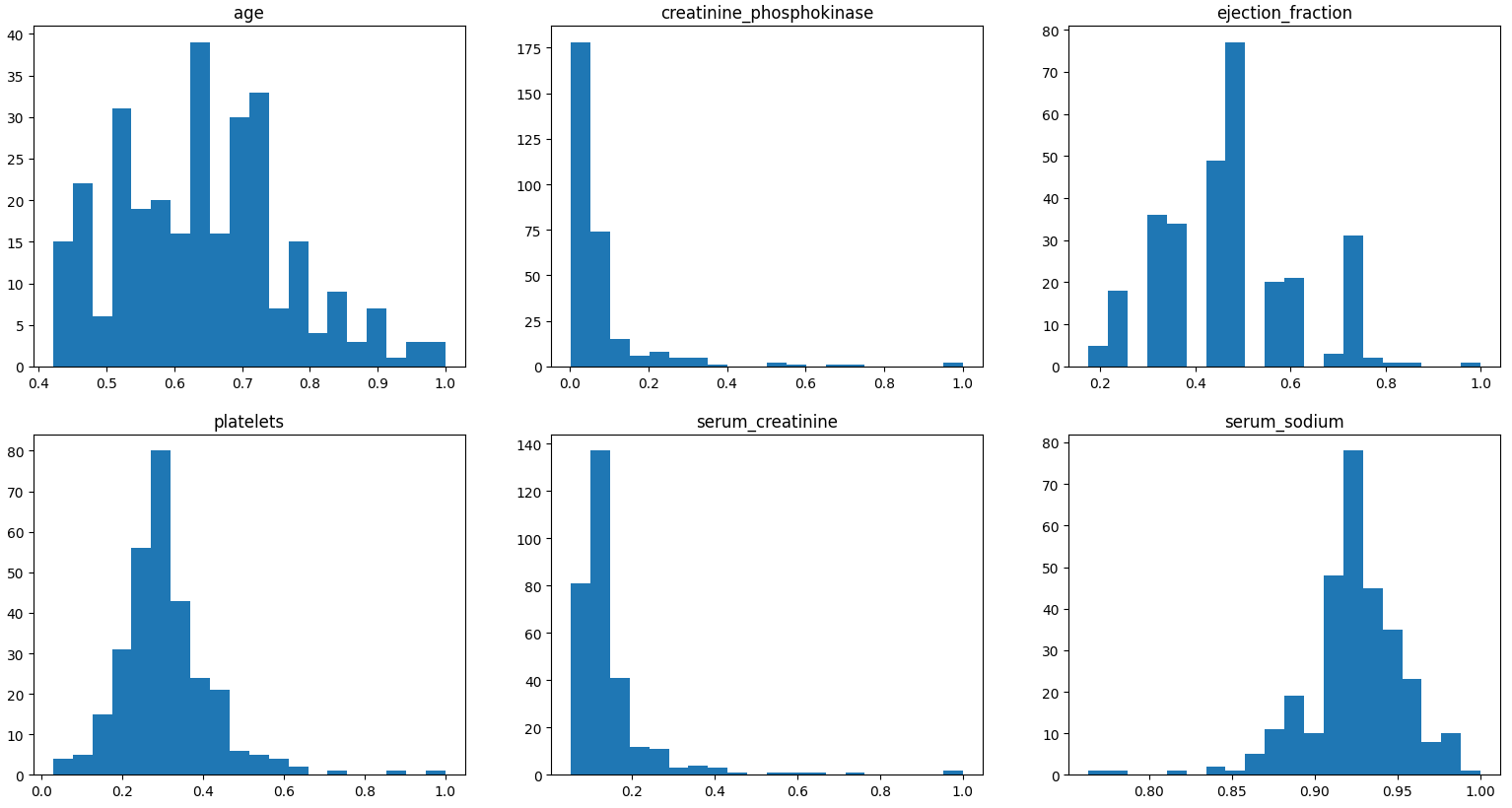
MinMaxScaler масштабирует данные к промежутку [0,1]

* 1. Через параметры MinMaxScaler определены минимальное и максимальное значения в данных для каждого признака.

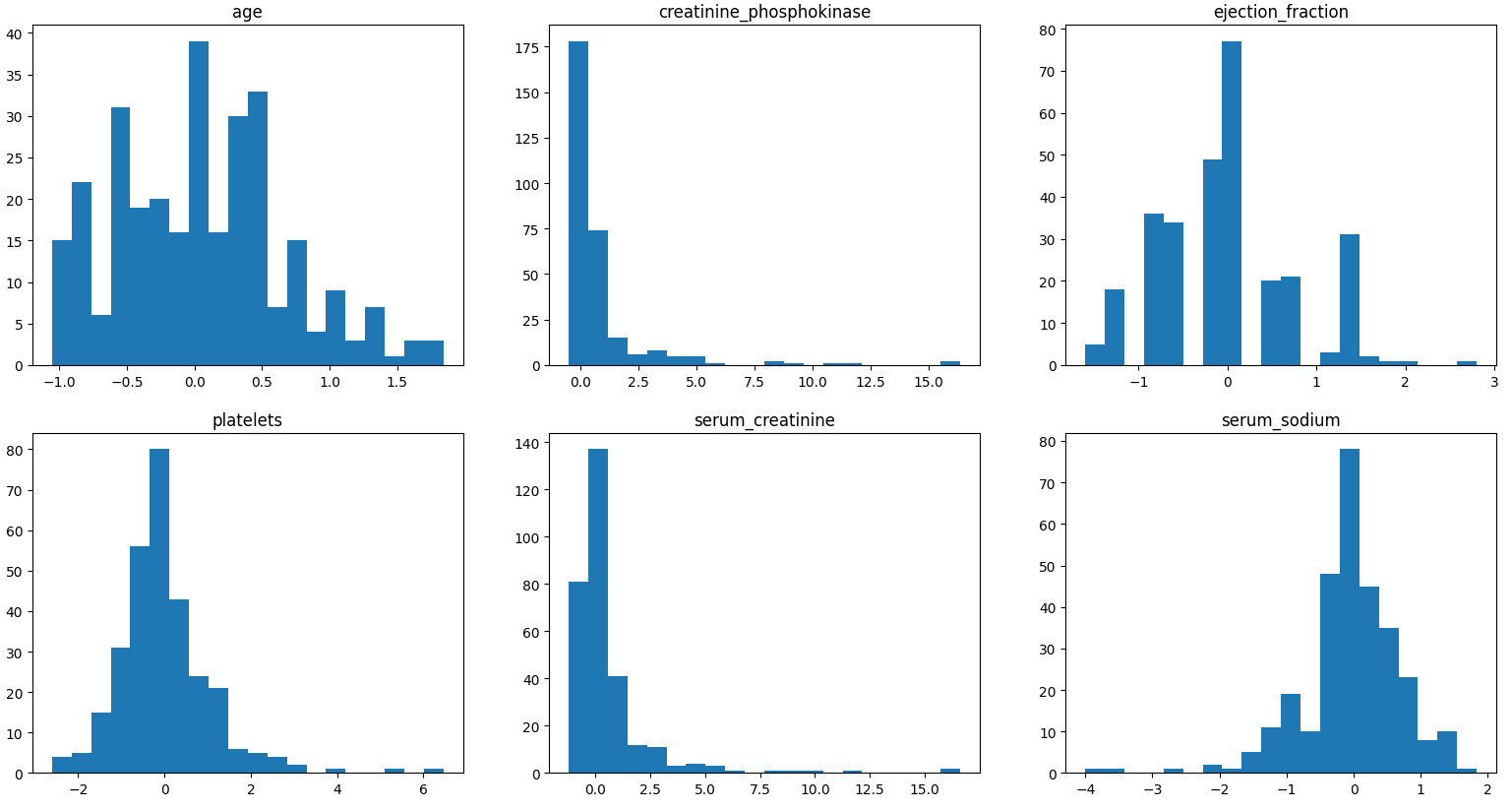


* 1. Трансформированы данные, используя MaxAbsScaler и RobustScaler. Построены гистограммы.

MaxAbsScaler:



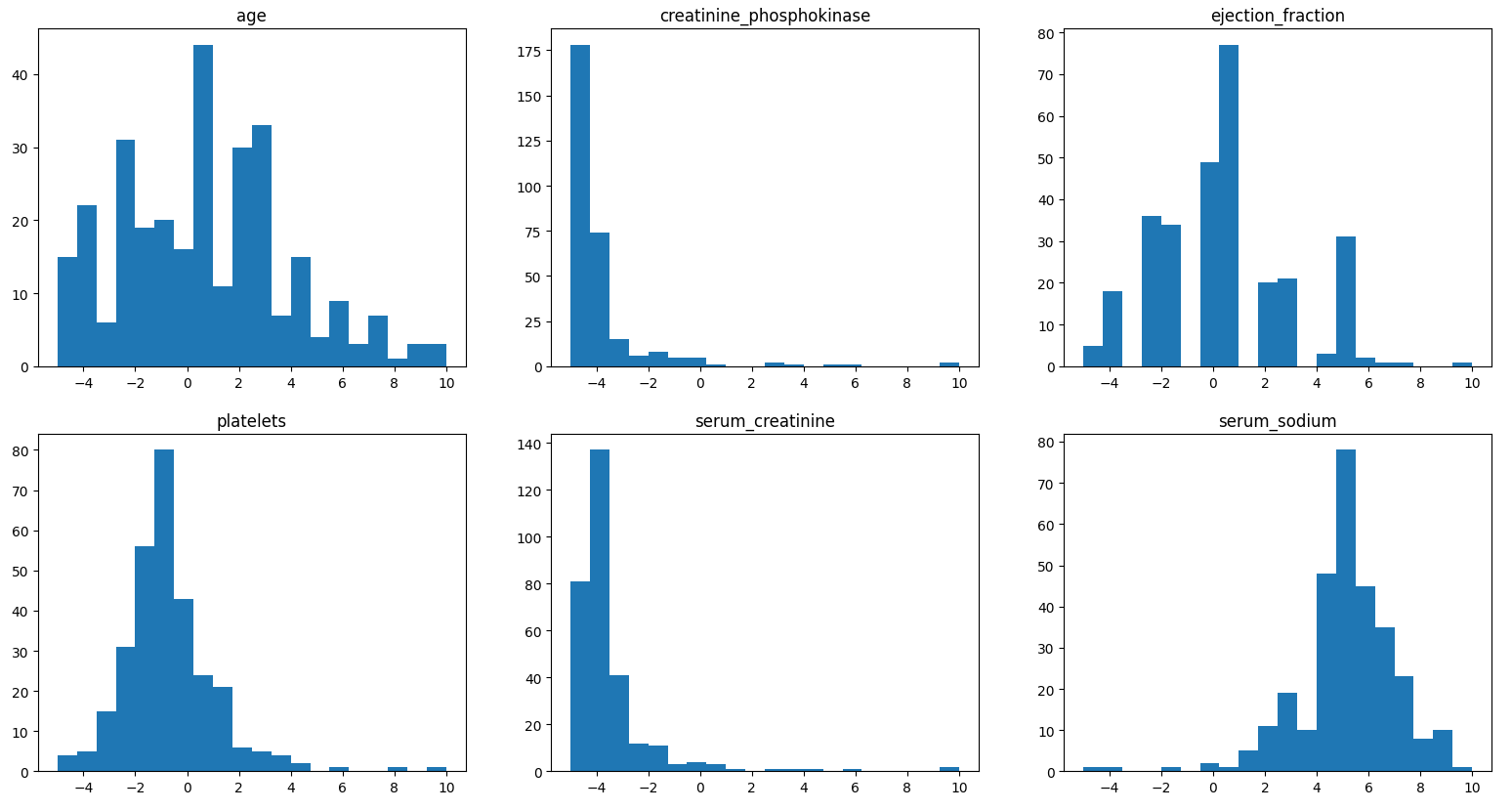
RobustScaler:



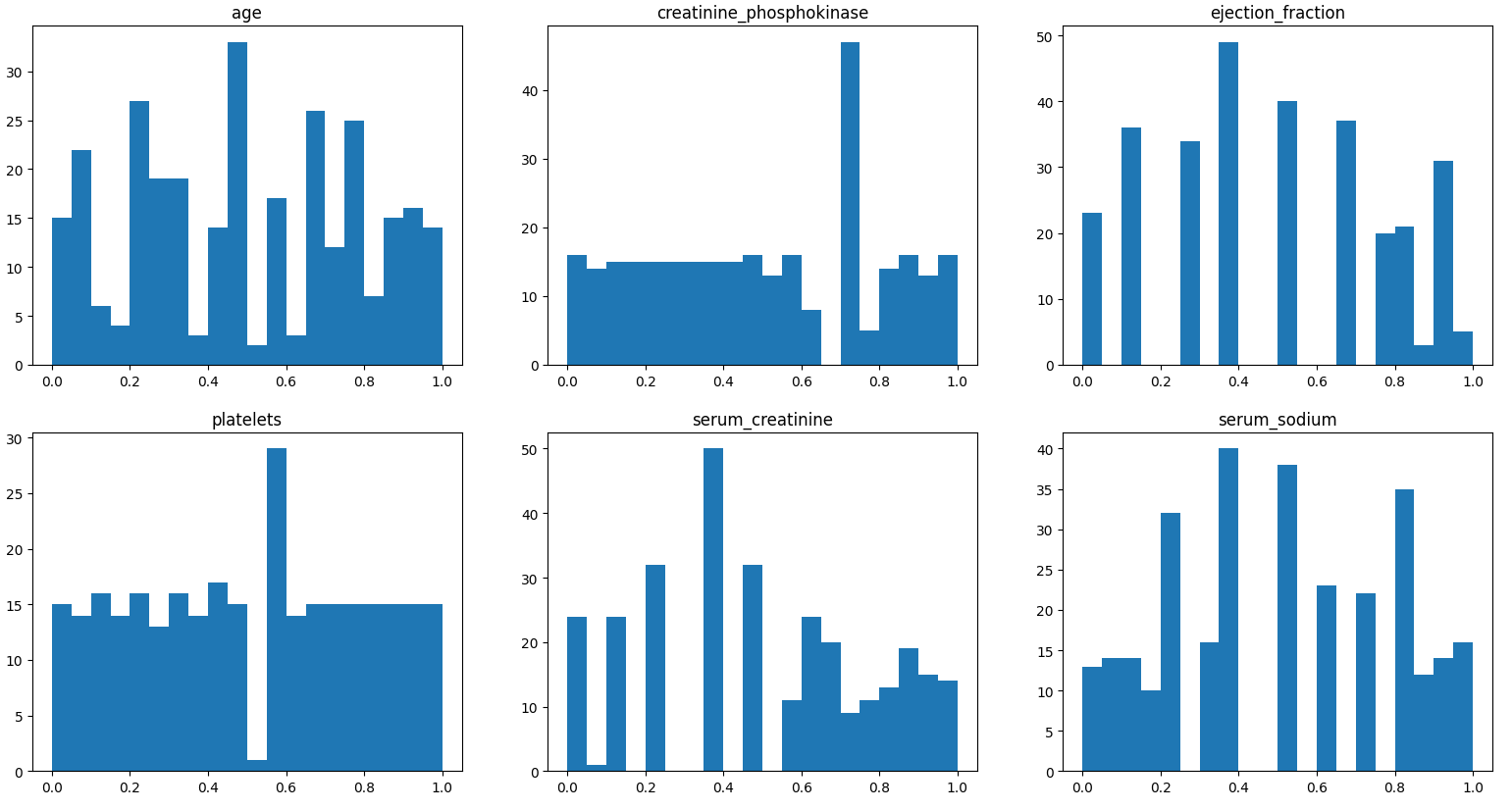
MaxAbsScaler изменяет данные таким образом, чтобы максимальное значение по модулю было равно 1. RobustScaler центрирует по медиане и масштабирует данные отбрасывая первые 25% и последние 25% диапазона.

* 1. Написана функция, которая приводит все данные к диапазону [-5 10].





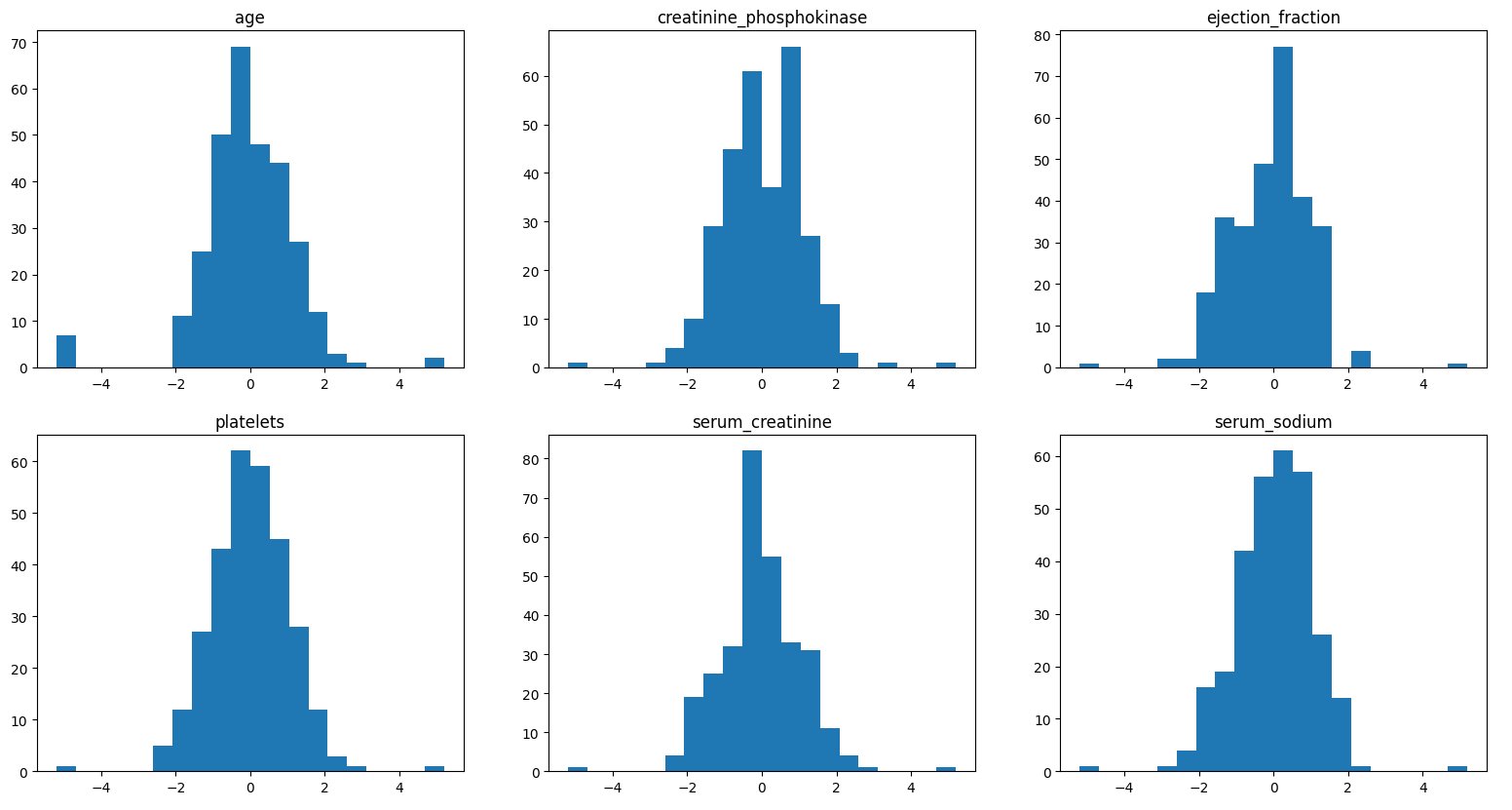
1. **Нелинейные преобразования**
   1. Данные приведены к равномерному распределению, используя QuantileTransformer. Построены диаграммы.



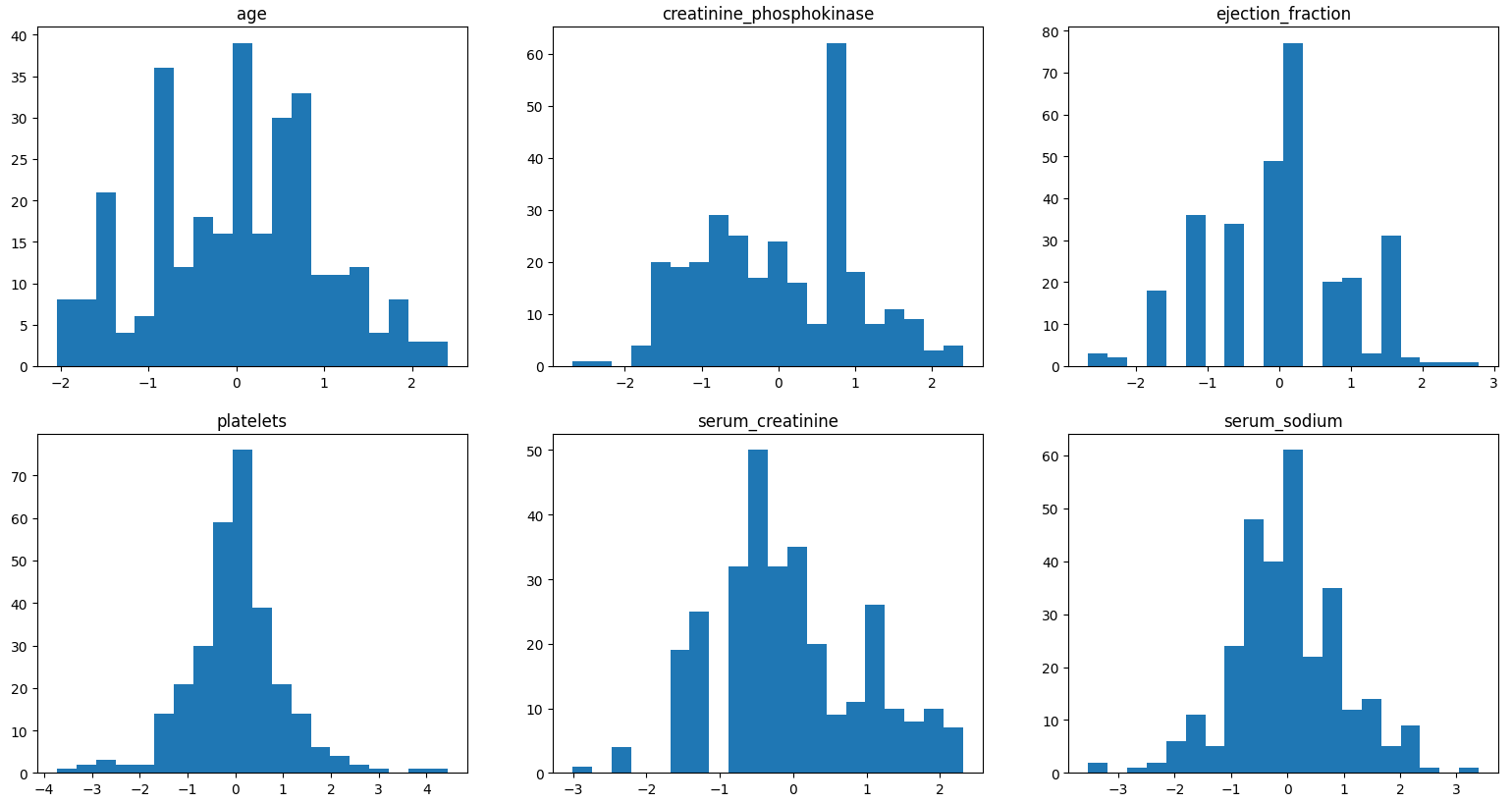
Равномерно распределяет, уменьшая пиковое количество наблюдений.

n\_quantiles влияет на точность вычисления функции распределения.

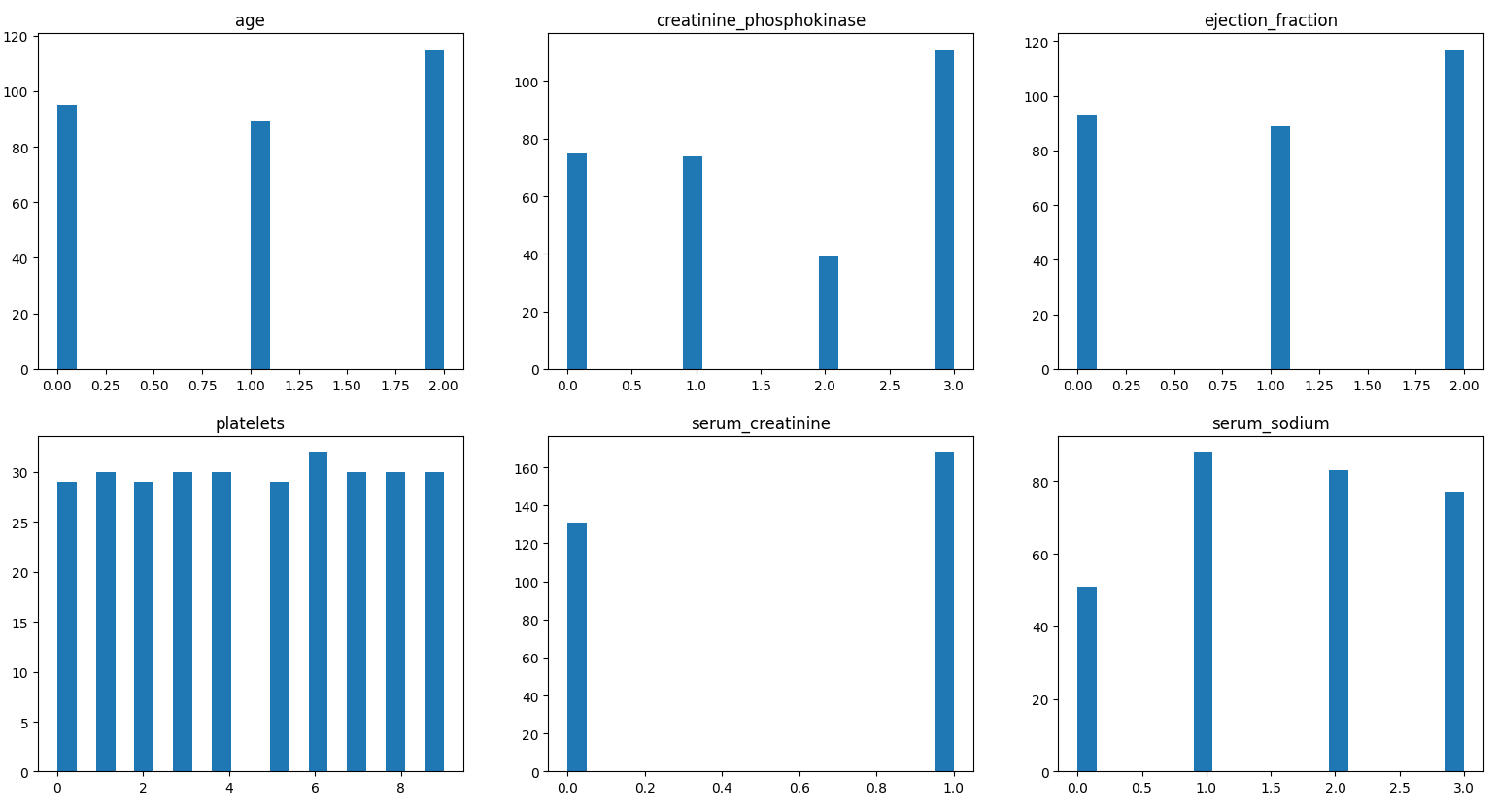
* 1. Данные приведены к нормальному распределению, передав в QuantileTransformer параметр output\_distribution=‘normal’



* 1. Данные приведены к нормальному распределению, используя PowerTransformer

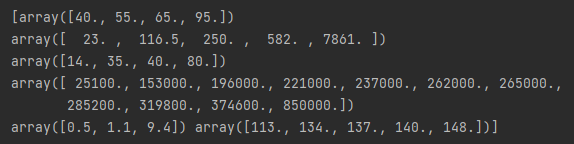


1. **Дискретизация признаков**
   1. Проведена дискретизация признаков, используя KBinsDiscretizer, на следующее количество диапазонов: age – 3, creatinine\_phosphokinase – 4, ejection\_fraction – 3, platelets – 10, serum\_creatinine – 2, serum\_sodium – 4.



Данные разбиваются на количество указанных диапазонов.

* 1. Через параметр bin\_edges\_ выведены диапазоны каждого интервала для каждого признака.



1. **Выводы**

Произведено знакомство с методами предобработки данных библиотеки Scikit Learn.

Выяснено, что стандартизация на основе не всех значений снижает качество выходных данных.

При приведении данных к диапазону гистограммы схожи между собой и отличаются диапазоном. При этом RobustScaler полезен против выбросов.

При нелинейных преобразованиях также выяснено, что QuantileTransform очень полезен против выбросов.

**ПРИЛОЖЕНИЕ А**

**Исходный код программы**

import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
from sklearn import preprocessing  
  
  
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)  
  
n\_bins = 20  
fig, axs = plt.subplots(2, 3)  
axs[0, 0].hist(df['age'].values, bins=n\_bins)  
axs[0, 0].set\_title('age')  
axs[0, 1].hist(df['creatinine\_phosphokinase'].values, bins=n\_bins)  
axs[0, 1].set\_title('creatinine\_phosphokinase')  
axs[0, 2].hist(df['ejection\_fraction'].values, bins=n\_bins)  
axs[0, 2].set\_title('ejection\_fraction')  
axs[1, 0].hist(df['platelets'].values, bins=n\_bins)  
axs[1, 0].set\_title('platelets')  
axs[1, 1].hist(df['serum\_creatinine'].values, bins=n\_bins)  
axs[1, 1].set\_title('serum\_creatinine')  
axs[1, 2].hist(df['serum\_sodium'].values, bins=n\_bins)  
axs[1, 2].set\_title('serum\_sodium')  
plt.show()  
  
  
def draw(data\_scaled):  
 fig, axs = plt.subplots(2, 3)  
 axs[0, 0].hist(data\_scaled[:, 0], bins=n\_bins)  
 axs[0, 0].set\_title('age')  
 axs[0, 1].hist(data\_scaled[:, 1], bins=n\_bins)  
 axs[0, 1].set\_title('creatinine\_phosphokinase')  
 axs[0, 2].hist(data\_scaled[:, 2], bins=n\_bins)  
 axs[0, 2].set\_title('ejection\_fraction')  
 axs[1, 0].hist(data\_scaled[:, 3], bins=n\_bins)  
 axs[1, 0].set\_title('platelets')  
 axs[1, 1].hist(data\_scaled[:, 4], bins=n\_bins)  
 axs[1, 1].set\_title('serum\_creatinine')  
 axs[1, 2].hist(data\_scaled[:, 5], bins=n\_bins)  
 axs[1, 2].set\_title('serum\_sodium')  
 plt.show()  
  
  
data = df.to\_numpy(dtype='float')  
  
print("Мат. ожидание до стандартизации:", [np.mean(i) for i in data.T])  
print("СКО до стандартизации:", [np.std(i) for i in data.T], "\n")  
  
scaler = preprocessing.StandardScaler().fit(data[:, :])  
  
print("Мат. ожидание поля mean\_:", scaler.mean\_)  
print("СКО поля var\_:", scaler.var\_, "\n")  
  
data\_scaled = scaler.transform(data)  
  
print("Мат. ожидание после стандартизации:", [np.mean(i) for i in data\_scaled.T])  
print("СКО после стандартизации:", [np.std(i) for i in data\_scaled.T], "\n")  
  
draw(data\_scaled)  
  
  
min\_max\_scaler = preprocessing.MinMaxScaler().fit(data)  
data\_min\_max\_scaled = min\_max\_scaler.transform(data)  
  
draw(data\_min\_max\_scaled)  
  
  
attributes = ['age', 'creatinine\_phosphokinase', 'ejection\_fraction', 'platelets', 'serum\_creatinine', 'serum\_sodium']  
for i in range(6):  
 print(attributes[i], "min:", min\_max\_scaler.data\_min\_[i], "max:", min\_max\_scaler.data\_max\_[i])  
  
max\_abs\_scaler = preprocessing.MaxAbsScaler().fit(data)  
max\_abs\_scaled = max\_abs\_scaler.transform(data)  
  
draw(max\_abs\_scaled)  
  
robust\_scaler = preprocessing.RobustScaler().fit(data)  
robust\_scaled = robust\_scaler.transform(data)  
  
draw(robust\_scaled)  
  
range\_scaler = preprocessing.MinMaxScaler(feature\_range=(-5, 10)).fit(data)  
range\_scaled = range\_scaler.transform(data)  
  
draw(range\_scaled)  
  
quantile\_transformer = preprocessing.QuantileTransformer(n\_quantiles=100, random\_state=0).fit(data)  
data\_quantile\_scaled = quantile\_transformer.transform(data)  
  
draw(data\_quantile\_scaled)  
  
quantile\_transformer = preprocessing.QuantileTransformer(n\_quantiles=100, random\_state=0, output\_distribution='normal').fit(data)  
data\_quantile\_scaled = quantile\_transformer.transform(data)  
  
draw(data\_quantile\_scaled)  
  
power\_transformer = preprocessing.PowerTransformer().fit(data)  
power\_scaled = power\_transformer.transform(data)  
  
draw(power\_scaled)  
  
disc\_transformer = preprocessing.KBinsDiscretizer(n\_bins=[3, 4, 3, 10, 2, 4], encode='ordinal').fit(data)  
disc\_scaled = disc\_transformer.transform(data)  
  
draw(disc\_scaled)  
print("\n", disc\_transformer.bin\_edges\_)