МИНОБРНАУКИ РОССИИ

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ

ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)

Кафедра МОЭВМ

ОТЧЕТ

по лабораторной работе №1

по дисциплине «Машинное обучение»

Тема: Предобработка данных

Студент гр. 6307 _____ Новиков Б.М.

Преподаватель _____ Жангиров Т.Р.

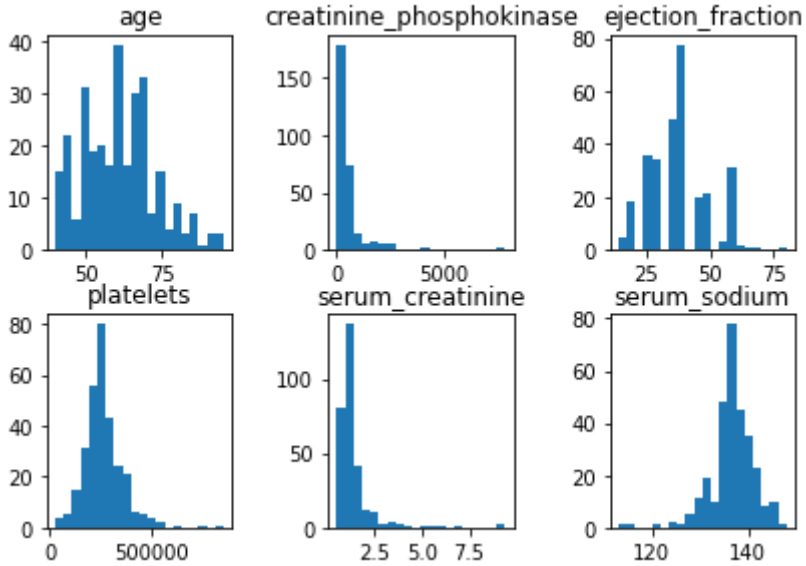
2020

Загрузка данных

1. Загрузить датасет с сайта kaggle.com, создать датфрейм на основе этого датасета и исключить бинарные признаки и признаки времени.

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

2. Построить гистограммы признаков.



4. Определить на основании гистограмм диапазоны значений для каждого признака и наиболее часто встречающееся значение.

```
diapason for age: (40.0 : 95.0)
diapason for creatinine_phosphokinase: (23 : 7861)
diapason for ejection_fraction: (14 : 80)
diapason for platelets: (25100.0 : 850000.0)
diapason for serum_creatinine: (0.5 : 9.4)
diapason for serum_sodium: (113 : 148)

most frequent value for age: 60.0
most frequent value for creatinine_phosphokinase: 582
most frequent value for ejection_fraction: 35
most frequent value for platelets: 263358.03
most frequent value for serum_creatinine: 1.0
most frequent value for serum_sodium: 136
```

5. Преобразовать датафрейм к двумерному массиву NumPy

```
data = df.to_numpy(dtype='float')
```

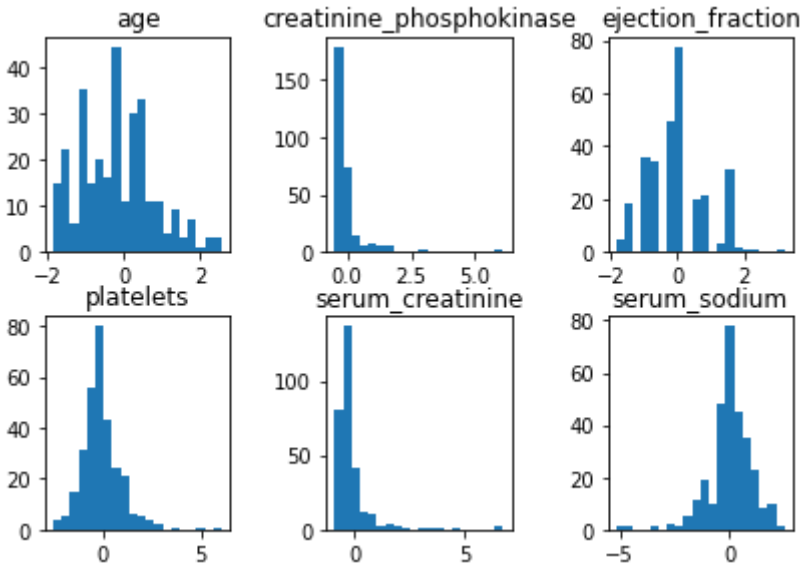
Стандартизация данных

1, 2. Подключить модуль Sklearn и настроить стандартизацию для первых 150 наблюдений, а также применить их к данным

```
from sklearn import preprocessing

scaler = preprocessing.StandardScaler().fit(data[:150,:])
data_scaled = scaler.transform(data)
```

3. Построить гистограммы стандартизированных данных.



4. Сравнить данные до стандартизации и после.

Стандартизация приводит все исходные значения набора данных, независимо от их начальных распределений и единиц измерения, к набору значений из распределения с нулевым средним и стандартным отклонением, равным 1. В результате формируется так называемая стандартизированная шкала, которая определяет место каждого значения в наборе данных, измеряя его отклонение от среднего в единицах стандартного отклонения.

5. Рассчитать мат. Ожидание и СКО до и после стандартизации. На основании этих значений вывести для каждого признака формулы, по которым они стандартизировались.

For age:

```
МО before стандарт.: 60.83389297658862
СКО before стандарт.: 11.874901429842655
МО after стандарт.: -0.16970362369106984
СКО after стандарт.: 0.9538237876978354

For creatinine_phosphokinase:
МО before стандарт.: 581.8394648829432
```

```
СКО before стандарт.: 968.6639668032415
МО after стандарт.: -0.021276750290383013
СКО after стандарт.: 0.8141790488228113

For ejection_fraction:
МО before стандарт.: 38.08361204013378
СКО before стандарт.: 11.815033462318585
МО after стандарт.: 0.01050249484809085
СКО after стандарт.: 0.9061082161919123

For platelets:
МО before стандарт.: 263358.02926421404
СКО before стандарт.: 97640.54765451424
МО after стандарт.: -0.035228788194085287
СКО after стандарт.: 1.0150611342848024

For serum_creatinine:
МО before стандарт.: 1.3938795986622072
СКО before стандарт.: 1.0327786652795918
МО after стандарт.: -0.10864080163893569
СКО after стандарт.: 0.8854288727548568

For serum_sodium:
МО before стандарт.: 136.62541806020067
СКО before стандарт.: 4.405092379513557
МО after стандарт.: 0.03790759894920013
СКО after стандарт.: 0.9703735961735016

Формулы для каждого признака: yi = (xi - M) / o

yi=(xi - 60.83389297658862) / 11.874901429842655
yi=(xi - 581.8394648829432) / 968.6639668032415
yi=(xi - 38.08361204013378) / 11.815033462318585
yi=(xi - 263358.02926421404) / 97640.54765451424
yi=(xi - 1.3938795986622072) / 1.0327786652795918
yi=(xi - 136.62541806020067) / 4.405092379513557
```

6. Сравнить значения из формул с полями mean_ и var_ объекта scaler.

```
For age:
МО: 62.946666666666665
Дисп: 154.99715555555557

For creatinine_phosphokinase:
МО: 607.1533333333333
Дисп: 1415488.8231555554

For ejection_fraction:
МО: 37.946666666666665
Дисп: 170.02382222222224

For platelets:
```

```
МО: 266746.74946666666
Дисп: 9252860499.078917

For serum_creatinine:
МО: 1.5206000000000002
Дисп: 1.3605269733333336

For serum_sodium:
МО: 136.45333333333335
Дисп: 20.607822222222225
```

Значения не совпадают, потому что формулы были выведены относительно всех наблюдений.

7. Провести настройку стандартизации на всех данных и сравнить с результатами настройки на основании 150 наблюдений.

```
For age:
МО before стандарт.: 60.83389297658862
СКО before стандарт.: 11.874901429842655
МО after стандарт.: 5.703353062957326e-16
СКО after стандарт.: 0.9999999999999998

For creatinine_phosphokinase:
МО before стандарт.: 581.8394648829432
СКО before стандарт.: 968.6639668032415
МО after стандарт.: 0.0
СКО after стандарт.: 1.0

For ejection_fraction:
МО before стандарт.: 38.08361204013378
СКО before стандарт.: 11.815033462318585
МО after стандарт.: -3.267546025652635e-17
СКО after стандарт.: 1.0

For platelets:
МО before стандарт.: 263358.02926421404
СКО before стандарт.: 97640.54765451424
МО after стандарт.: 7.723290606088045e-17
СКО after стандарт.: 1.0

For serum_creatinine:
МО before стандарт.: 1.3938795986622072
СКО before стандарт.: 1.0327786652795918
МО after стандарт.: 1.4258382657393315e-16
СКО after стандарт.: 1.0

For serum_sodium:
МО before стандарт.: 136.62541806020067
СКО before стандарт.: 4.405092379513557
МО after стандарт.: -8.673849449914267e-16
СКО after стандарт.: 0.9999999999999999
```

For age:

```
MO: 60.83389297658862
Дисп: 141.01328396847913

For creatinine_phosphokinase:
MO: 581.8394648829432
Дисп: 938309.8805829913

For ejection_fraction:
MO: 38.08361204013378
Дисп: 139.5950157157079

For platelets:
MO: 263358.02926421404
Дисп: 9533676546.273466

For serum_creatinine:
MO: 1.3938795986622072
Дисп: 1.066631771456695

For serum_sodium:
MO: 136.62541806020067
Дисп: 19.404838872048412
```

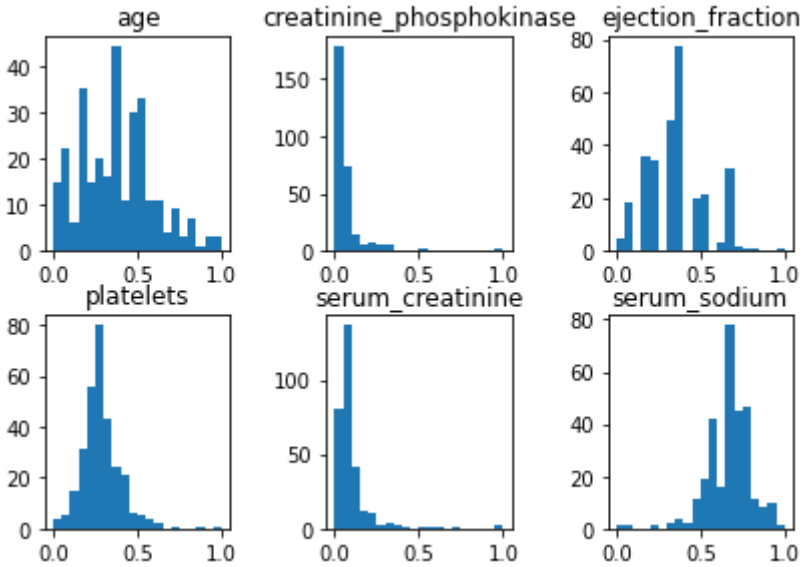
Теперь параметры стандартизации совпадают.

Приведение к диапазону.

1. Привести данные к диапазону, используя MinMaxScaler

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

2. Построить гистограммы для признаков и сравнить



Данные оказались преведены к диапозону от 0 до 1.

3. Через параметры MinMaxScaler определить минимальное и максимальное значение в данных для каждого признака.

min_max_scaler.data_max_

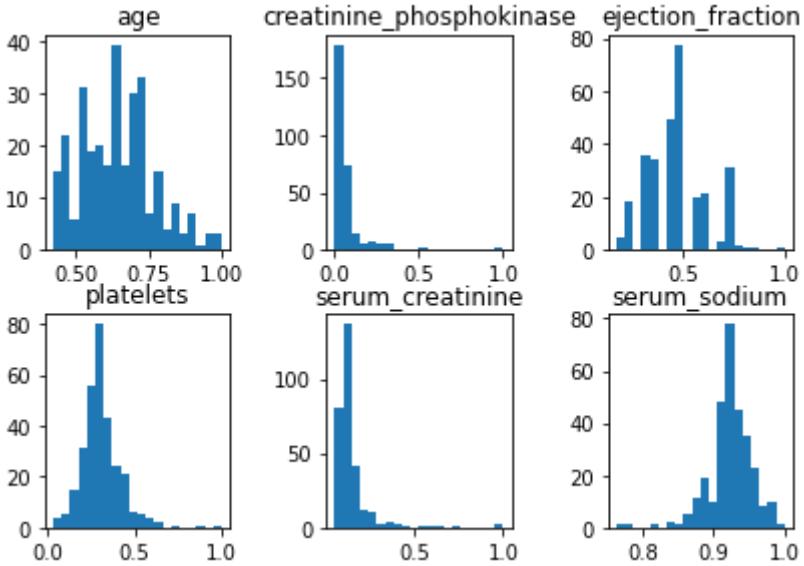
array([9.500e+01, 7.861e+03, 8.000e+01, 8.500e+05, 9.400e+00, 1.480e+02])

min_max_scaler.data_min_

array([4.00e+01, 2.30e+01, 1.40e+01, 2.51e+04, 5.00e-01, 1.13e+02])

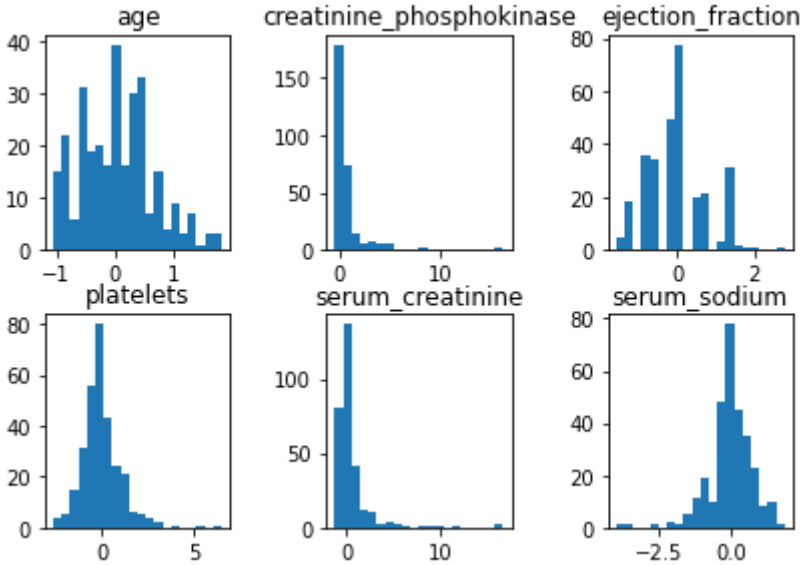
4. Аналогично трасформируйте данные, используя MaxAbsScaler и RobustScaler.

```
max_abs_scaler = preprocessing.MaxAbsScaler().fit(data)
data_max_abs_scaled = max_abs_scaler.transform(data)
```



Значения наблюдений для каждого признака делятся на максимальное значение наблюдения. Диапозон от 0 до 1.

```
robust_scaler = preprocessing.RobustScaler().fit(data)
data_robust_scaled = robust_scaler.transform(data)
```



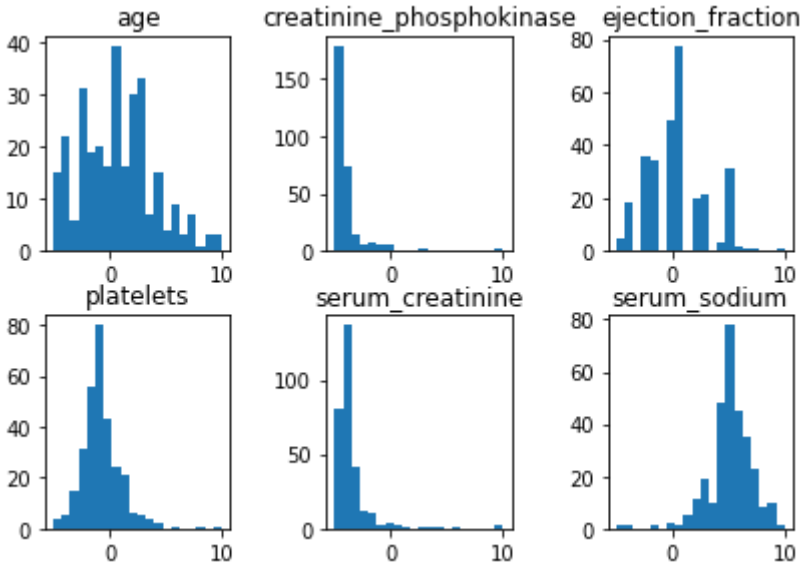
Из значений наблюдений для каждого признака удаляется медианной значение, а после они приводятся межквартильному диапозону.

5. Написать функцию, которая приводит все данные к диапазону [-5 10]

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

yi = (xi — xmin) / (xmax — xmin) * (10 + 5) - 5

yi = 15 * (xi — xmin) / (xmax - xmin) — 5

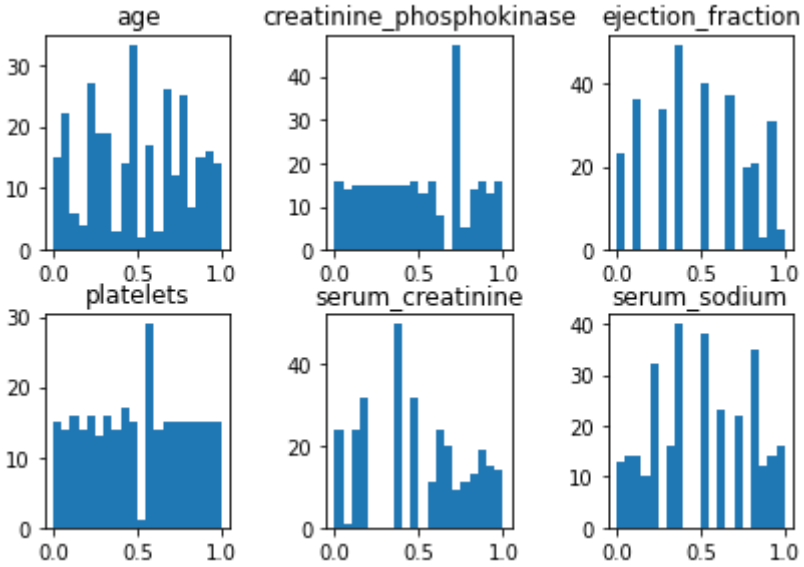


Нелинейные преобразования.

1. Привести данные к равномерному распределению, используя QuantileTransformer

```
quantile_transformer = preprocessing.QuantileTransformer(n_quantiles = 100,random_state=0).fit(data)
data_quantile_scaled = quantile_transformer.transform(data)
```

2. Построить гистограммы и сравнить с исходными данными



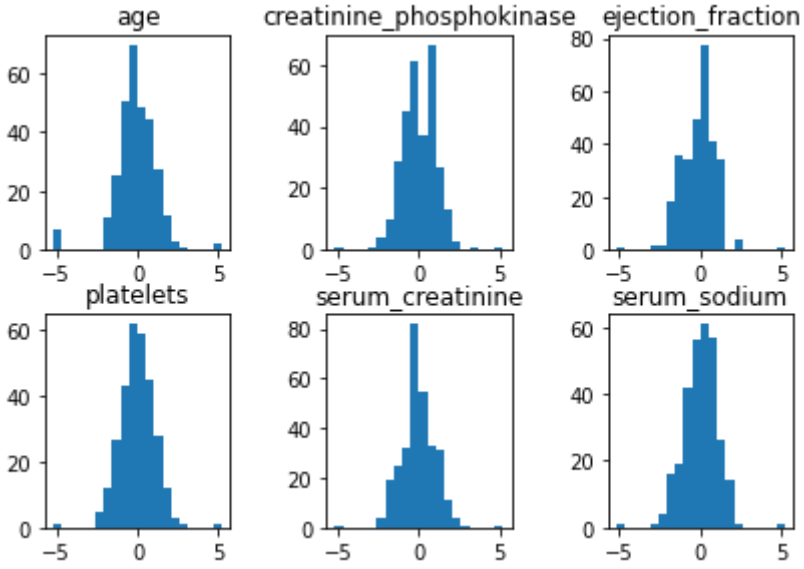
3. Определите как и на что влияет значение параметра n_quantiles

`n_quantiles` влияет на то, насколько точно приведутся исходные данные к равномерному распределению. Данное значение используется для задания размера шага при дискретизации оценочной CDF для исходных данных.

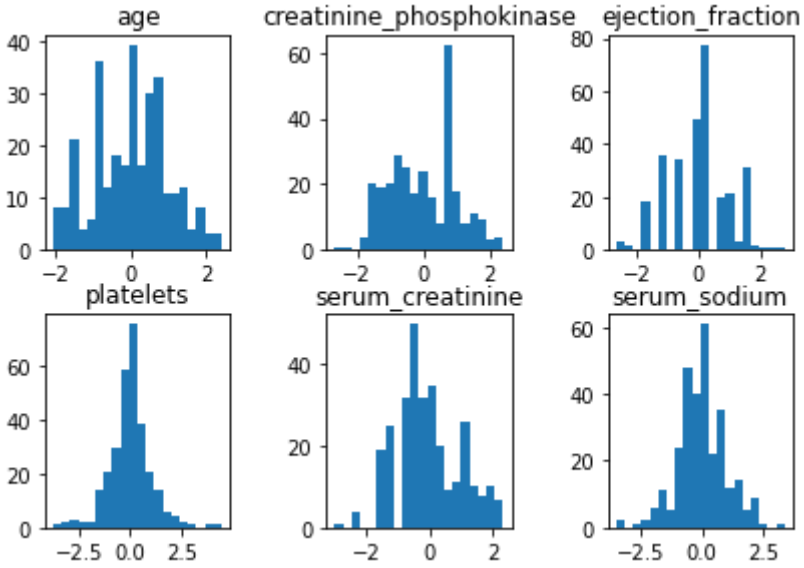
4. Приведите данные к нормальному распределению, передав параметр output_distribution="normal"

```
quantile_transformer = preprocessing.QuantileTransformer(n_quantiles = 100,random_state=0, output_distribution="normal").fit(data)
data_quantile_scaled = quantile_transformer.transform(data)
```

5. Построить гистограммы и сравнить с исходными данными



6. Самостоятельно привидите данные к нормальному распределению, используя PowerTransformer



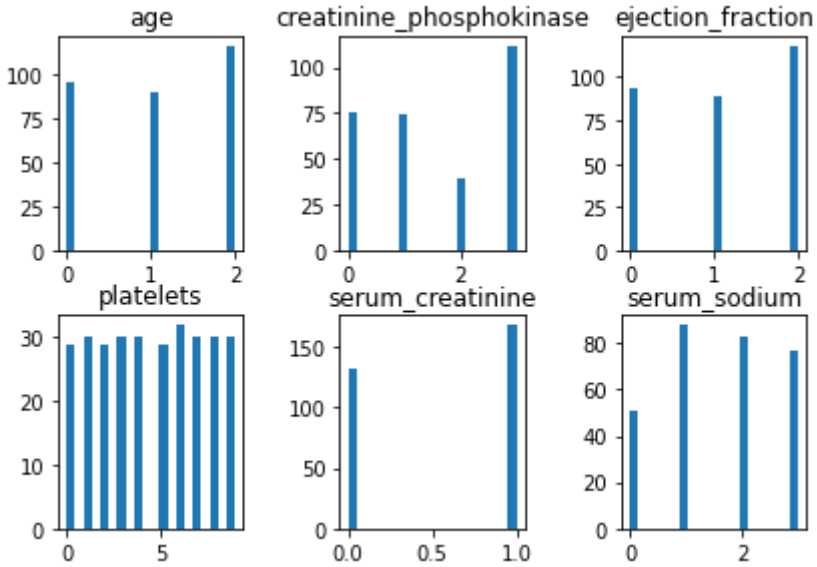
Дискретизация признаков

1. Проведите дискретизацию признаков, используя KBinsDiscretizer, на следующее количество диапазонов:

age — 3
creatinine_phosphokinase — 4
ejection_fraction — 3
platelets — 10
serum_creatinine — 2
serum_sodium — 4

```
bins = [3, 4, 3, 10, 2, 4]
discretizer = preprocessing.KBinsDiscretizer(n_bins=bins, encode='ordinal')
discrete_data = discretizer.fit_transform((data), 'discr')
```

2. Построить гистрограммы. Объяснить результаты



Данные дискретизированы на диапозоны.

3. Через параметр bin_edges_ выведите диапазоны каждого интервала для каждого признака

```
discretizer.bin_edges_
```

```
array([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.])],
      dtype=object)
```