РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра информационных технологий

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 3

Дисциплина: Методы машинного обучения

Студент: Турсунов Баходурхон Азимджонович

Группа: НФИбд-03-19

Москва 2022

```
import numpy as np import
tensorflow as tf import
tensorflow_datasets as tfds
import numpy as np
from sklearn.metrics import
mean_absolute_error import matplotlib.pyplot
as plt from mpl_toolkits import mplot3d
import pandas as pd
```

Вариант № 26

In [1]:

1. Загрузите заданный в индивидуальномзадании набор данных из Tensorflow Datasets, включая указанные в задании независимый

```
ds = tfds.load("wine_quality", split='train') data1 = tfds.as_dataframe(ds)
data = data1[['features/total sulfur dioxide',
'features/chlorides']].dropna()
```

признак и зависимый признак (отклик).

In [2]:

2. Решите задачу полиномиальнойрегрессии для степени полинома, указанной в индивидуальном задании, при помощи нейронной сети с одним нейроном и оцените качество полученной модели по показателю, указанному в

```
import numpy as np
_x = data['features/total sulfur dioxide'].values
X2 = np.hstack([_x.reshape(-1,1), _x.reshape(-1,1)**2,_x.reshape(-
1,1)**3,_x.resha reg2_model =
tf.keras.Sequential([tf.keras.Input(shape=(4,)),tf.keras.layers.Dens
reg2_model.compile(loss='mean_absolute_error')
history = reg2_model.fit(X2, data['features/chlorides'].values,
epochs=200,verbos
```

индивидуальном задании.

In [3]:

e

e

```
Epoch 1/200
108/108 [==============================] - 1s 5ms/step - loss: 620421952.0000 -
va l_loss: 592523328.0000
Epoch 2/200
108/108 [==============================] - 0s 3ms/step - loss: 557468672.0000 -
va l_loss: 530310816.0000 Epoch 3/200
108/108 [==============================] - 0s 2ms/step - loss: 495059936.0000 -
va l_loss: 466274656.0000
Epoch 4/200
108/108 [==============================] - 0s 2ms/step - loss: 431696096.0000 -
va l_loss: 402784224.0000
Epoch 5/200
108/108 [==============================] - 0s 2ms/step - loss: 369593280.0000 -
va l_loss: 340749632.0000
Epoch 6/200
108/108 [==============================] - 0s 2ms/step - loss: 306434592.0000 -
va l_loss: 277904032.0000 Epoch 7/200
108/108 [==============================] - 0s 2ms/step - loss: 244325488.0000 -
va l_loss: 214213440.0000
Epoch 8/200
108/108 [==============================] - 0s 2ms/step - loss: 183629024.0000 -
va l_loss: 152815696.0000
Epoch 9/200
108/108 [==============================] - 0s 2ms/step - loss: 119183944.0000 -
va l_loss: 88721888.0000
Epoch 10/200
108/108 [==============================] - 0s 2ms/step - loss: 58604496.0000 -
val
_loss: 25847338.0000
Epoch 11/200
108/108 [==============================] - 0s 2ms/step - loss: 5618178.0000 -
val_ loss: 569448.8125
Epoch 12/200
108/108 [==============================] - 0s 2ms/step - loss: 316987.2188 -
val_l oss: 704929.6250
Epoch 13/200
108/108 [==============================] - 0s 2ms/step - loss: 329939.0938 -
val_l oss: 730701.7500
Epoch 14/200
108/108 [==============================] - 0s 3ms/step - loss: 313482.6875 -
val_l oss: 576960.4375 Epoch 15/200
108/108 [==============================] - 0s 2ms/step - loss: 316624.2188 -
val_l oss: 368526.9688
Epoch 16/200
108/108 [==============================] - 0s 2ms/step - loss: 302842.4375 -
val_l oss: 198983.9531
Epoch 17/200
108/108 [==============================] - 0s 2ms/step - loss: 313666.7188 -
val_l oss: 327740.8750
Epoch 18/200
108/108 [==============================] - 0s 2ms/step - loss: 300126.2500 -
val_l oss: 198952.7500 Epoch 19/200
108/108 [==============================] - 0s 2ms/step - loss: 305992.9062 -
val_l oss: 539164.3125
Epoch 20/200
108/108 [==============================] - 0s 2ms/step - loss: 307724.8125 -
val_l oss: 374406.3438
Epoch 21/200
108/108 [==============================] - 0s 2ms/step - loss: 306854.6250 -
val_l oss: 919196.8125 Epoch 22/200
```

```
108/108 [==============================] - 0s 2ms/step - loss: 293361.8438 -
val_l oss: 289463.4062 Epoch 23/200
108/108 [==============================] - 0s 2ms/step - loss: 296435.7188 -
val_l oss: 124042.0469
Epoch 24/200
108/108 [==============================] - 0s 2ms/step - loss: 305930.3750 -
val_l oss: 719624.1250
Epoch 25/200
108/108 [==============================] - 0s 2ms/step - loss: 298573.4375 -
val_l oss: 64677.0234
Epoch 26/200
108/108 [==============================] - 0s 3ms/step - loss: 303743.5000 -
val_l oss: 271645.6562 Epoch 27/200
108/108 [==============================] - 0s 2ms/step - loss: 298892.7500 -
val_l oss: 150085.1094
Epoch 28/200
108/108 [==============================] - 0s 3ms/step - loss: 311559.3438 -
val_l oss: 478984.7500
Epoch 29/200
108/108 [==============================] - 0s 3ms/step - loss: 301789.5625 -
val_l oss: 542344.8125
Epoch 30/200
108/108 [==============================] - 0s 3ms/step - loss: 307842.9375 -
val_l oss: 142651.3750 Epoch 31/200
108/108 [==============================] - 0s 2ms/step - loss: 289877.6562 -
val_l oss: 1182938.5000
Epoch 32/200
108/108 [==============================] - 0s 2ms/step - loss: 309583.5938 -
val_l oss: 73338.6484
Epoch 33/200
108/108 [==============================] - 0s 2ms/step - loss: 305969.9688 -
val_l oss: 859024.8750
Epoch 34/200
108/108 [==============================] - 0s 2ms/step - loss: 303571.3125 -
val_l oss: 198862.6250 Epoch 35/200
108/108 [==============================] - 0s 2ms/step - loss: 305494.9062 -
val_l oss: 904066.2500
Epoch 36/200
108/108 [==============================] - 0s 2ms/step - loss: 301293.7812 -
val_l oss: 341938.1250
Epoch 37/200
108/108 [==============================] - 0s 2ms/step - loss: 298260.0625 -
val_l oss: 197906.2656
Epoch 38/200
108/108 [==============================] - 0s 3ms/step - loss: 303986.8125 -
val_l oss: 702559.5625 Epoch 39/200
108/108 [==============================] - 0s 2ms/step - loss: 308780.6250 -
val_l oss: 576931.5625
Epoch 40/200
108/108 [==============================] - 0s 3ms/step - loss: 304494.5000 -
val_l oss: 540288.8750
Epoch 41/200
108/108 [==============================] - 0s 3ms/step - loss: 294284.3438 -
val_l oss: 171153.1250
Epoch 42/200
108/108 [==============================] - 0s 3ms/step - loss: 307808.8438 -
val_l oss: 661561.3125 Epoch 43/200
108/108 [==============================] - 0s 3ms/step - loss: 310941.7188 -
val_l oss: 172565.3281
Epoch 44/200
```

```
108/108 [==============================] - 0s 4ms/step - loss: 296833.2812 -
val_l oss: 880780.6875
Epoch 45/200
108/108 [==============================] - 0s 3ms/step - loss: 306906.5625 -
val_l oss: 438412.4062
Epoch 46/200
108/108 [==============================] - 0s 3ms/step - loss: 305932.5312 -
val_l oss: 809020.6875 Epoch 47/200
108/108 [==============================] - 0s 3ms/step - loss: 302283.5625 -
val_l oss: 217046.0312
Epoch 48/200
108/108 [==============================] - 0s 2ms/step - loss: 308737.5312 -
val_l oss: 252342.7031
Epoch 49/200
108/108 [==============================] - 0s 3ms/step - loss: 302061.0625 -
val_l oss: 837259.7500
Epoch 50/200
108/108 [==============================] - 0s 4ms/step - loss: 312683.7500 -
val_l oss: 74737.6719 Epoch 51/200
108/108 [==============================] - 0s 4ms/step - loss: 306794.3750 -
val_l oss: 119747.2422
Epoch 52/200
108/108 [==============================] - 0s 4ms/step - loss: 302227.0625 -
val_l oss: 147097.3750
Epoch 53/200
108/108 [==============================] - 0s 4ms/step - loss: 308423.8438 -
val_l oss: 31127.2266
Epoch 54/200
108/108 [==============================] - 0s 3ms/step - loss: 307839.1562 -
val_l oss: 156337.8125 Epoch 55/200
108/108 [==============================] - 0s 3ms/step - loss: 309784.3750 -
val_l oss: 106698.1875
Epoch 56/200
108/108 [==============================] - 0s 3ms/step - loss: 303454.4375 -
val_l oss: 123179.5078
Epoch 57/200
108/108 [==============================] - 0s 4ms/step - loss: 306782.5625 -
val_l oss: 452328.2188
Epoch 58/200
108/108 [==============================] - 0s 3ms/step - loss: 311414.3438 -
val_l oss: 663518.0000 Epoch 59/200
108/108 [==============================] - 0s 3ms/step - loss: 308067.8750 -
val_l oss: 87014.4609
Epoch 60/200
108/108 [==============================] - 0s 3ms/step - loss: 308955.7188 -
val_l oss: 245135.9062
Epoch 61/200
108/108 [==============================] - 0s 3ms/step - loss: 312906.2812 -
val_l oss: 14745.2607
Epoch 62/200
108/108 [==============================] - 0s 4ms/step - loss: 298353.3750 -
val_l oss: 145434.2031 Epoch 63/200
108/108 [==============================] - 0s 3ms/step - loss: 306915.1250 -
val_l oss: 309906.5938
Epoch 64/200
108/108 [==============================] - 0s 4ms/step - loss: 311976.4375 -
val_l oss: 398815.3750
Epoch 65/200
108/108 [==============================] - 0s 3ms/step - loss: 309299.5938 -
val_l oss: 532569.1875
```

```
Epoch 66/200
108/108 [==============================] - 0s 3ms/step - loss: 307251.7812 -
val_l oss: 139582.9062 Epoch 67/200
108/108 [==============================] - 0s 3ms/step - loss: 316214.8438 -
val_l oss: 231341.1406
Epoch 68/200
108/108 [==============================] - 0s 3ms/step - loss: 299616.8125 -
val_l oss: 15553.7949
Epoch 69/200
108/108 [==============================] - 0s 3ms/step - loss: 305842.4062 -
val_l oss: 257380.7188
Epoch 70/200
108/108 [==============================] - 0s 2ms/step - loss: 311517.4375 -
val_l oss: 647174.6875 Epoch 71/200
108/108 [==============================] - 0s 3ms/step - loss: 305676.2812 -
val_l oss: 80852.6641
Epoch 72/200
108/108 [==============================] - 0s 3ms/step - loss: 310127.6875 -
val_l oss: 1325100.0000
Epoch 73/200
108/108 [==============================] - 0s 2ms/step - loss: 314359.6875 -
val_l oss: 206507.7500
Epoch 74/200
108/108 [==============================] - 0s 2ms/step - loss: 310827.0938 -
val_l oss: 319853.3750 Epoch 75/200
108/108 [==============================] - 0s 2ms/step - loss: 311326.7812 -
val_l oss: 9191.7637
Epoch 76/200
108/108 [==============================] - 0s 2ms/step - loss: 300589.9375 -
val_l oss: 128358.3984
Epoch 77/200
108/108 [==============================] - 0s 2ms/step - loss: 312180.3438 -
val_l oss: 59138.9883
Epoch 78/200
108/108 [==============================] - 0s 3ms/step - loss: 306583.9688 -
val_l oss: 686866.3125 Epoch 79/200
108/108 [==============================] - 0s 2ms/step - loss: 310114.4375 -
val_l oss: 293691.0938
Epoch 80/200
108/108 [==============================] - 0s 3ms/step - loss: 310756.9062 -
val_l oss: 785444.7500
Epoch 81/200
108/108 [==============================] - 0s 2ms/step - loss: 307370.2500 -
val_l oss: 151897.1250
Epoch 82/200
108/108 [==============================] - 0s 2ms/step - loss: 308522.0312 -
val_l oss: 715037.9375 Epoch 83/200
108/108 [==============================] - 0s 2ms/step - loss: 312055.0938 -
val_l oss: 348025.8125
Epoch 84/200
108/108 [==============================] - 0s 2ms/step - loss: 307017.7500 -
val_l oss: 225550.7969
Epoch 85/200
108/108 [==============================] - 0s 2ms/step - loss: 306378.7812 -
val_l oss: 119232.8359 Epoch 86/200
108/108 [==============================] - 0s 3ms/step - loss: 308623.3125 -
val_l oss: 747349.6250 Epoch 87/200
108/108 [==============================] - 0s 3ms/step - loss: 311608.9375 -
val_l oss: 354883.4688
Epoch 88/200
```

```
108/108 [==============================] - 0s 3ms/step - loss: 311153.9688 -
val_l oss: 758165.5000
Epoch 89/200
108/108 [==============================] - 0s 3ms/step - loss: 311139.6875 -
val_l oss: 442053.5312
Epoch 90/200
108/108 [==============================] - 0s 3ms/step - loss: 308439.8125 -
val_l oss: 70339.1875 Epoch 91/200
108/108 [==============================] - 0s 3ms/step - loss: 312297.6250 -
val_l oss: 295262.2500
Epoch 92/200
108/108 [==============================] - 0s 3ms/step - loss: 307526.8125 -
val_l oss: 35491.2422
Epoch 93/200
108/108 [==============================] - 0s 3ms/step - loss: 311077.8125 -
val_l oss: 793971.5625
Epoch 94/200
108/108 [==============================] - 0s 3ms/step - loss: 311233.4375 -
val_l oss: 354128.3125 Epoch 95/200
108/108 [==============================] - 0s 3ms/step - loss: 306552.6250 -
val_l oss: 265276.8750
Epoch 96/200
108/108 [==============================] - 0s 3ms/step - loss: 310327.5000 -
val_l oss: 938505.1250
Epoch 97/200
108/108 [==============================] - 0s 3ms/step - loss: 308756.4375 -
val_l oss: 28349.2227
Epoch 98/200
108/108 [==============================] - 0s 3ms/step - loss: 306786.9688 -
val_l oss: 76055.6719 Epoch 99/200
108/108 [==============================] - 0s 3ms/step - loss: 307496.5938 -
val_l oss: 158846.4531
Epoch 100/200
108/108 [==============================] - 0s 3ms/step - loss: 309989.0312 -
val_l oss: 112814.7266
Epoch 101/200
108/108 [==============================] - 0s 3ms/step - loss: 309338.4688 -
val_l oss: 256141.5469
Epoch 102/200
108/108 [==============================] - 0s 3ms/step - loss: 302680.7500 -
val_l oss: 66443.4531 Epoch 103/200
108/108 [==============================] - 0s 3ms/step - loss: 309176.0000 -
val_l oss: 216322.0312
Epoch 104/200
108/108 [==============================] - 0s 2ms/step - loss: 309147.3438 -
val_l oss: 12605.9971
Epoch 105/200
108/108 [==============================] - 0s 2ms/step - loss: 306123.7500 -
val_l oss: 36895.5742
Epoch 106/200
108/108 [==============================] - 0s 2ms/step - loss: 307939.5000 -
val_l oss: 352425.5000 Epoch 107/200
108/108 [==============================] - 0s 2ms/step - loss: 314854.1250 -
val_l oss: 108721.8281
Epoch 108/200
108/108 [==============================] - 0s 3ms/step - loss: 305904.4375 -
val_l oss: 472672.5625
Epoch 109/200
108/108 [==============================] - 0s 3ms/step - loss: 307957.0000 -
val_l oss: 315156.1250
```

```
Epoch 110/200
108/108 [==============================] - 0s 2ms/step - loss: 308593.4062 -
val_l oss: 249549.7969 Epoch 111/200
108/108 [==============================] - 0s 3ms/step - loss: 309778.8125 -
val_l oss: 14009.5615
Epoch 112/200
108/108 [==============================] - 0s 2ms/step - loss: 311745.7812 -
val_l oss: 85981.2969
Epoch 113/200
108/108 [==============================] - 0s 3ms/step - loss: 304995.1875 -
val_l oss: 278746.2812
Epoch 114/200
108/108 [==============================] - 0s 3ms/step - loss: 310442.3750 -
val_l oss: 298358.3125 Epoch 115/200
108/108 [==============================] - 0s 2ms/step - loss: 316458.5000 -
val_l oss: 240982.4375
Epoch 116/200
108/108 [==============================] - 0s 3ms/step - loss: 304488.3125 -
val_l oss: 356243.2500
Epoch 117/200
108/108 [==============================] - 0s 3ms/step - loss: 314893.7188 -
val_l oss: 550165.6250
Epoch 118/200
108/108 [==============================] - 0s 3ms/step - loss: 308590.1250 -
val_l oss: 407653.5312 Epoch 119/200
108/108 [==============================] - 0s 3ms/step - loss: 306667.3438 -
val_l oss: 453452.5000
Epoch 120/200
108/108 [==============================] - 0s 3ms/step - loss: 306632.2188 -
val_l oss: 539897.0000
Epoch 121/200
108/108 [==============================] - 0s 3ms/step - loss: 310656.2812 -
val_l oss: 124615.5859
Epoch 122/200
108/108 [==============================] - 0s 3ms/step - loss: 310466.8125 -
val_l oss: 406085.9688 Epoch 123/200
108/108 [==============================] - 0s 3ms/step - loss: 305643.1875 -
val_l oss: 594728.0625
Epoch 124/200
108/108 [==============================] - 0s 3ms/step - loss: 311286.1562 -
val_l oss: 125484.5078
Epoch 125/200
108/108 [==============================] - 0s 3ms/step - loss: 304742.9375 -
val_l oss: 340619.3125
Epoch 126/200
108/108 [==============================] - 0s 3ms/step - loss: 311597.4375 -
val_l oss: 357076.6875 Epoch 127/200
108/108 [==============================] - 0s 3ms/step - loss: 311863.0312 -
val_l oss: 105764.8984
Epoch 128/200
108/108 [==============================] - 0s 3ms/step - loss: 309553.6250 -
val_l oss: 110113.1406
Epoch 129/200
108/108 [==============================] - 0s 2ms/step - loss: 308186.5000 -
val_l oss: 35576.3281
Epoch 130/200
108/108 [==============================] - 0s 2ms/step - loss: 311175.7812 -
val_l oss: 450618.6562 Epoch 131/200
108/108 [==============================] - 0s 3ms/step - loss: 315754.5312 -
val_l oss: 258835.0312
```

```
Epoch 132/200
108/108 [==============================] - 0s 3ms/step - loss: 304764.0000 -
val_l oss: 518769.5312
Epoch 133/200
108/108 [==============================] - 0s 3ms/step - loss: 305569.8438 -
val_l oss: 1063417.0000
Epoch 134/200
108/108 [==============================] - 0s 3ms/step - loss: 314206.8125 -
val_l oss: 68237.6406 Epoch 135/200
108/108 [==============================] - 0s 2ms/step - loss: 308410.5625 -
val_l oss: 671723.0625
Epoch 136/200
108/108 [==============================] - 0s 2ms/step - loss: 311724.1875 -
val_l oss: 172255.8750
Epoch 137/200
108/108 [==============================] - 0s 2ms/step - loss: 305408.3750 -
val_l oss: 161771.3750
Epoch 138/200
108/108 [==============================] - 0s 2ms/step - loss: 312307.9375 -
val_l oss: 39568.7930 Epoch 139/200
108/108 [==============================] - 0s 2ms/step - loss: 308895.1562 -
val_l oss: 412773.8438
Epoch 140/200
108/108 [==============================] - 0s 2ms/step - loss: 305278.7188 -
val_l oss: 129372.4062
Epoch 141/200
108/108 [==============================] - 0s 2ms/step - loss: 311173.3438 -
val_l oss: 403301.6250
Epoch 142/200
108/108 [==============================] - 0s 2ms/step - loss: 308336.1562 -
val_l oss: 385613.8750 Epoch 143/200
108/108 [==============================] - 0s 2ms/step - loss: 308603.1562 -
val_l oss: 218429.0469
Epoch 144/200
108/108 [==============================] - 0s 2ms/step - loss: 313567.0312 -
val_l oss: 447152.3125
Epoch 145/200
108/108 [==============================] - 0s 3ms/step - loss: 311543.5312 -
val_l oss: 375268.9688
Epoch 146/200
108/108 [==============================] - 0s 3ms/step - loss: 307876.3438 -
val_l oss: 96523.7031 Epoch 147/200
108/108 [==============================] - 0s 3ms/step - loss: 310382.7500 -
val_l oss: 863343.3750
Epoch 148/200
108/108 [==============================] - 0s 3ms/step - loss: 316107.3438 -
val_l oss: 132494.2656
Epoch 149/200
108/108 [==============================] - 0s 2ms/step - loss: 312218.2188 -
val_l oss: 331234.1250 Epoch 150/200
108/108 [==============================] - 0s 3ms/step - loss: 304692.4375 -
val_l oss: 230871.3594 Epoch 151/200
108/108 [==============================] - 0s 3ms/step - loss: 307496.0000 -
val_l oss: 216414.9531
Epoch 152/200
108/108 [==============================] - 0s 2ms/step - loss: 307569.5938 -
val_l oss: 228811.1875
Epoch 153/200
108/108 [==============================] - 0s 3ms/step - loss: 307894.3438 -
val_l oss: 285088.2812
```

```
Epoch 154/200
108/108 [==============================] - 0s 2ms/step - loss: 309451.3125 -
val_l oss: 260934.9688 Epoch 155/200
108/108 [==============================] - 0s 2ms/step - loss: 310576.6250 -
val_l oss: 75274.5078
Epoch 156/200
108/108 [==============================] - 0s 2ms/step - loss: 308789.3438 -
val_l oss: 115518.1094
Epoch 157/200
108/108 [==============================] - 0s 2ms/step - loss: 312606.3125 -
val_l oss: 49083.7969
Epoch 158/200
108/108 [==============================] - 0s 2ms/step - loss: 312905.1875 -
val_l oss: 305359.0000 Epoch 159/200
108/108 [==============================] - 0s 2ms/step - loss: 307490.9688 -
val_l oss: 569195.9375
Epoch 160/200
108/108 [==============================] - 0s 2ms/step - loss: 312721.9062 -
val_l oss: 39505.9180
Epoch 161/200
108/108 [==============================] - 0s 2ms/step - loss: 304591.3750 -
val_l oss: 107604.5859
Epoch 162/200
108/108 [==============================] - 0s 2ms/step - loss: 307932.8125 -
val_l oss: 129717.6797 Epoch 163/200
108/108 [==============================] - 0s 2ms/step - loss: 310822.6562 -
val_l oss: 58197.5664
Epoch 164/200
108/108 [==============================] - 0s 2ms/step - loss: 315582.8438 -
val_l oss: 488134.7500
Epoch 165/200
108/108 [==============================] - 0s 2ms/step - loss: 302336.7812 -
val_l oss: 366654.7500
Epoch 166/200
108/108 [==============================] - 0s 2ms/step - loss: 309886.5000 -
val_l oss: 284741.5312 Epoch 167/200
108/108 [==============================] - 0s 2ms/step - loss: 303943.4062 -
val_l oss: 196582.2500
Epoch 168/200
108/108 [==============================] - 0s 2ms/step - loss: 308323.5938 -
val_l oss: 1022293.0625
Epoch 169/200
108/108 [==============================] - 0s 2ms/step - loss: 313884.3438 -
val_l oss: 109192.8594
Epoch 170/200
108/108 [==============================] - 0s 3ms/step - loss: 309238.6250 -
val_l oss: 15477.9941 Epoch 171/200
108/108 [==============================] - 0s 2ms/step - loss: 309421.6562 -
val_l oss: 77382.8047
Epoch 172/200
108/108 [==============================] - 0s 2ms/step - loss: 305886.5625 -
val_l oss: 261766.4219
Epoch 173/200
108/108 [==============================] - 0s 2ms/step - loss: 307698.3438 -
val_l oss: 355296.4688
Epoch 174/200
108/108 [==============================] - 0s 3ms/step - loss: 311085.3750 -
val_l oss: 119265.0469 Epoch 175/200
108/108 [==============================] - 0s 2ms/step - loss: 306255.7500 -
val_l oss: 490926.3125
```

```
Epoch 176/200
108/108 [==============================] - 0s 2ms/step - loss: 309619.2188 -
val_l oss: 24113.2812
Epoch 177/200
108/108 [==============================] - 0s 2ms/step - loss: 309563.6250 -
val_l oss: 923432.7500
Epoch 178/200
108/108 [==============================] - 0s 3ms/step - loss: 312320.4375 -
val_l oss: 213590.1094 Epoch 179/200
108/108 [==============================] - 0s 2ms/step - loss: 312498.5312 -
val_l oss: 616999.1250
Epoch 180/200
108/108 [==============================] - 0s 2ms/step - loss: 310592.3438 -
val_l oss: 154436.4375
Epoch 181/200
108/108 [==============================] - 0s 3ms/step - loss: 310798.3750 -
val_l oss: 31849.0664
Epoch 182/200
108/108 [==============================] - 0s 2ms/step - loss: 306119.9375 -
val_l oss: 173076.7656 Epoch 183/200
108/108 [==============================] - 0s 2ms/step - loss: 308203.0312 -
val_l oss: 36356.8633
Epoch 184/200
108/108 [==============================] - 0s 3ms/step - loss: 310220.2188 -
val_l oss: 312257.5938
Epoch 185/200
108/108 [==============================] - 0s 2ms/step - loss: 309409.3125 -
val_l oss: 549374.3750
Epoch 186/200
108/108 [==============================] - 0s 2ms/step - loss: 307167.1875 -
val_l oss: 481173.1875 Epoch 187/200
108/108 [==============================] - 0s 2ms/step - loss: 311257.1562 -
val_l oss: 540251.5000
Epoch 188/200
108/108 [==============================] - 0s 2ms/step - loss: 312421.8750 -
val_l oss: 15007.0449
Epoch 189/200
108/108 [==============================] - 0s 3ms/step - loss: 308355.5938 -
val_l oss: 301426.4688
Epoch 190/200
108/108 [==============================] - 0s 2ms/step - loss: 307521.8750 -
val_l oss: 869998.2500 Epoch 191/200
108/108 [==============================] - 0s 2ms/step - loss: 313490.4688 -
val_l oss: 851762.5000
Epoch 192/200
108/108 [==============================] - 0s 2ms/step - loss: 310495.5938 -
val_l oss: 601981.5625
Epoch 193/200
108/108 [==============================] - 0s 2ms/step - loss: 309271.7500 -
val_l oss: 1242969.6250
Epoch 194/200
108/108 [==============================] - 0s 2ms/step - loss: 309596.0312 -
val_l oss: 73525.5625 Epoch 195/200
108/108 [==============================] - 0s 2ms/step - loss: 313189.5625 -
val_l oss: 307649.1562
Epoch 196/200
108/108 [==============================] - 0s 2ms/step - loss: 310955.7500 -
val_l oss: 116410.9531
Epoch 197/200
```

```
108/108 [==============================] - 0s 2ms/step - loss: 308898.4375 -
val_l oss: 126391.3047
Epoch 198/200
108/108 [==============================] - 0s 2ms/step - loss: 307936.9375 -
val_l oss: 50382.9609 Epoch 199/200
108/108 [==============================] - 0s 2ms/step - loss: 307622.1562 -
val_l oss: 328546.1250
Epoch 200/200
108/108 [==============================] - 0s 2ms/step - loss: 302100.4688 -
val_l

oss: 492458.8125
```

In [4]:
```
mean_absolute_error(data['features/chlorides'],
 reg2_model.predict(X2))
```

```
489786.6
```

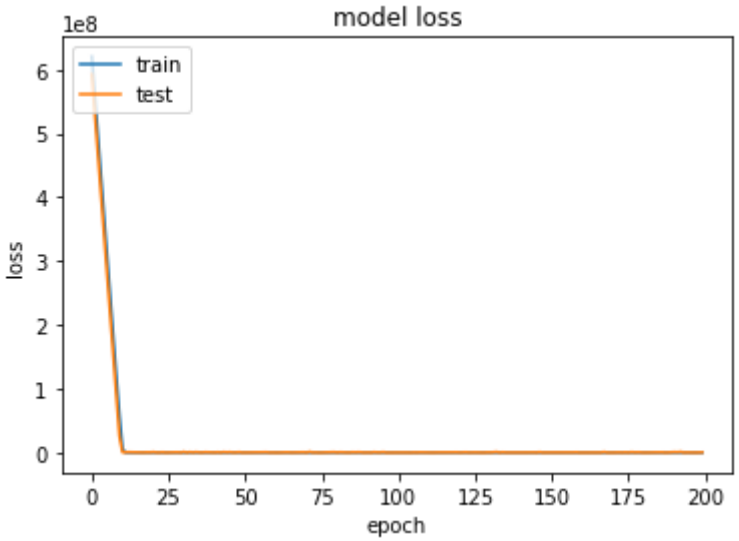
Out[4]:

3. Постройте кривые обучения

```
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss') plt.ylabel('loss')
plt.xlabel('epoch') plt.legend(['train',
'test'], loc='upper left') plt.show()
```

сзависимостью от количества эпох.

In [5]:

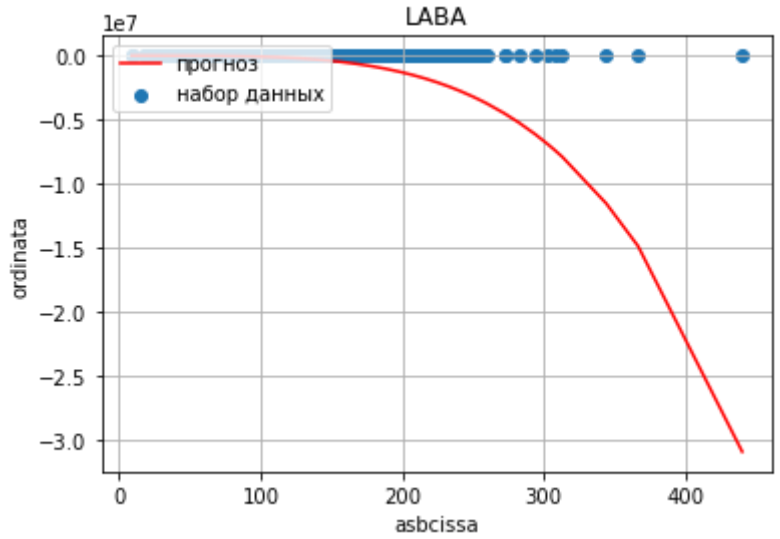


4. Визуализируйте точки набора данныхна плоскости в виде диаграммы рассеяния (ось X – независимый признак, ось Y – зависимый признак), а также линию регрессии (другим

```
plt.title('LABA')
plt.xlabel('asbcissa')
plt.ylabel('ordinata')
plt.scatter(_x, data['features/chlorides'], label='набор данных')
plt.plot(np.sort(_x), reg2_model.predict(X2)[np.argsort(_x)], color='red',
label=' plt.legend(loc='upper left') plt.grid();
```

цветом), подписывая оси и рисунок.

In [6]:



5. Определите в исходном набореданных признак (отличный от независимого и зависимого признаков), принимающий непрерывные значения и имеющий

свойства, указанные в индивидуальном

```
data1.std()
```

задании.

In [7]:

Out[7]:
```
features/alcohol                 1.230623
features/chlorides               0.021848
features/citric acid             0.121020
features/density                 0.002991
features/fixed acidity           0.843867
features/free sulfur dioxide    17.007133
features/pH                      0.151001
features/residual sugar          5.072068
features/sulphates               0.114126
features/total sulfur dioxide   42.498055
features/volatile acidity        0.100794
quality                          0.885639
dtype: float64
```

6. Визуализируйте этот признак всоответствии с индивидуальным

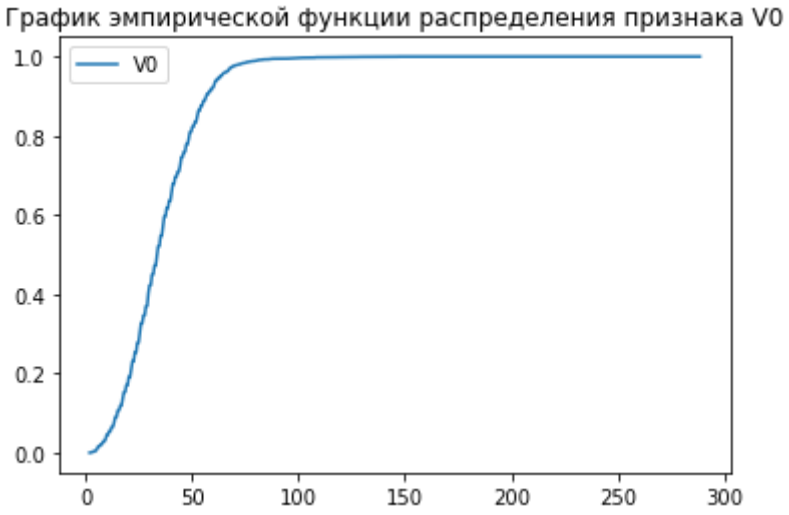
```
def ECDF(data, x):
counter = 0
for v in data:
if v <= x:
            counter += 1     return counter / len(data)
samples = data1['features/free sulfur dioxide'] # sepal
length npoints = 500
dx = (samples.max()-samples.min())/npoints

xlist = [samples.min()+dx*i for i in range(npoints)]
ylist = [ECDF(samples, x) for x in xlist]

df_ECDF = pd.DataFrame(ylist, columns=['V0'],index=xlist)
df_ECDF
df_ECDF.plot.line(title='График эмпирической функции распределения признака
V0');
```

заданием.

In [8]:



7. Сформируйте набор входных из двухпризнаков набора данных (независимый признак и определенный признак), создайте и адаптируйте нормализующий слой Tensorflow для

```
data2=data1[['features/total sulfur dioxide', 'features/free sulfur dioxide']]
feature_normalizer = tf.keras.layers.Normalization(axis=None,input_shape=(2,))
feature_normalizer.adapt(data2)
```

двух признаков.

In [9]:

8. Используя созданный нормализующийслой, постройте нейронную сеть (нелинейный регресор) с количеством скрытых слоев, количеством нейронов и функцией активации, указанными в индивидуальном задании, и одним нейроном в выходном слое и обучите

ее на наборе данных из двух признаков

```
large_model = tf.keras.Sequential([feature_normalizer,
                                   tf.keras.layers.Dense(units=64,
activation='ta
tf.keras.layers.Dense(units=64, activation='ta
tf.keras.layers.Dense(units=64, activation='ta
tf.keras.layers.Dense(units=64, activation='ta
tf.keras.layers.Dense(units=1)]) large_model.summary()
large_model.compile(loss='mae') history = large_model.fit(data2,
data1['features/chlorides'], epochs=200,verbose=
```

и отклика.

In [10]:

n
n
n
n

1

```
Model: "sequential_1"
_________________________________________________________________
_  Layer (type)                 Output Shape              Param #
=================================================================
=  normalization (Normalizatio  (None, 2)                3
n)
dense_1 (Dense)              (None, 64)                192
dense_2 (Dense)              (None, 64)                4160
dense_3 (Dense)              (None, 64)                4160
dense_4 (Dense)              (None, 64)                4160
dense_5 (Dense)              (None, 1)
65
=================================================================
Total params: 12,740
Trainable params: 12,737
Non-trainable params: 3
_________________________________________________________________
Epoch 1/200
108/108 [==============================] - 1s 4ms/step - loss: 0.0662 -
val_loss:
0.0341
Epoch 2/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0171 -
val_loss:
0.0123
Epoch 3/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0158 -
val_loss:
0.0114
Epoch 4/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0144 -
val_loss:
0.0116
Epoch 5/200
```

```
108/108 [==============================] - 0s 3ms/step - loss: 0.0146 -
val_loss:
0.0141
Epoch 6/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0136 -
val_loss:
0.0129
Epoch 7/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0135 -
val_loss:
0.0122
Epoch 8/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0132 -
val_loss:
0.0110
Epoch 9/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0128 -
val_loss:
0.0138
Epoch 10/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0127 -
val_loss:
0.0195
Epoch 11/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0123 -
val_loss:
0.0187
Epoch 12/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0123 -
val_loss:
0.0158
Epoch 13/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0121 -
val_loss:
0.0106
Epoch 14/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0115 -
val_loss:
0.0107
Epoch 15/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0118 -
val_loss:
0.0105
Epoch 16/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0115 -
val_loss:
0.0118
Epoch 17/200
108/108 [==============================] - 0s 4ms/step - loss: 0.0115 -
val_loss:
0.0136
Epoch 18/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0112 -
val_loss:
0.0112
Epoch 19/200
108/108 [==============================] - 0s 4ms/step - loss: 0.0112 -
val_loss:
0.0158
Epoch 20/200
```

```
108/108 [==============================] - 0s 3ms/step - loss: 0.0113 -
val_loss:
0.0111
Epoch 21/200
108/108 [==============================] - 0s 4ms/step - loss: 0.0111 -
val_loss:
0.0154
Epoch 22/200
108/108 [==============================] - 0s 4ms/step - loss: 0.0111 -
val_loss:
0.0107
Epoch 23/200
108/108 [==============================] - 0s 4ms/step - loss: 0.0111 -
val_loss:
0.0115
Epoch 24/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0110 -
val_loss:
0.0125
Epoch 25/200
108/108 [==============================] - 0s 4ms/step - loss: 0.0109 -
val_loss:
0.0105
Epoch 26/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0109 -
val_loss:
0.0146
Epoch 27/200
108/108 [==============================] - 0s 4ms/step - loss: 0.0109 -
val_loss:
0.0151
Epoch 28/200
108/108 [==============================] - 0s 4ms/step - loss: 0.0109 -
val_loss:
0.0110
Epoch 29/200
108/108 [==============================] - 0s 4ms/step - loss: 0.0108 -
val_loss:
0.0120
Epoch 30/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0108 -
val_loss:
0.0106
Epoch 31/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0108 -
val_loss:
0.0116
Epoch 32/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0108 -
val_loss:
0.0116
Epoch 33/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0107 -
val_loss:
0.0120
Epoch 34/200
108/108 [==============================] - 0s 4ms/step - loss: 0.0108 -
val_loss:
0.0128
Epoch 35/200
```

```
108/108 [==============================] - 1s 7ms/step - loss: 0.0108 -
val_loss:
0.0106
Epoch 36/200
108/108 [==============================] - 0s 4ms/step - loss: 0.0107 -
val_loss:
0.0124
Epoch 37/200
108/108 [==============================] - 0s 4ms/step - loss: 0.0107 -
val_loss:
0.0138
Epoch 38/200
108/108 [==============================] - 0s 4ms/step - loss: 0.0107 -
val_loss:
0.0116
Epoch 39/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0107 -
val_loss:
0.0113
Epoch 40/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0107 -
val_loss:
0.0107
Epoch 41/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0107 -
val_loss:
0.0107
Epoch 42/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0106 -
val_loss:
0.0110
Epoch 43/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0107 -
val_loss:
0.0108
Epoch 44/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0106 -
val_loss:
0.0105
Epoch 45/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0106 -
val_loss:
0.0104
Epoch 46/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0106 -
val_loss:
0.0108
Epoch 47/200
108/108 [==============================] - 0s 4ms/step - loss: 0.0106 -
val_loss:
0.0106
Epoch 48/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0106 -
val_loss:
0.0119
Epoch 49/200
108/108 [==============================] - 0s 4ms/step - loss: 0.0106 -
val_loss:
0.0106
Epoch 50/200
```

```
108/108 [==============================] - 0s 4ms/step - loss: 0.0106 -
val_loss:
0.0104
Epoch 51/200
108/108 [==============================] - 0s 4ms/step - loss: 0.0106 -
val_loss:
0.0112
Epoch 52/200
108/108 [==============================] - 0s 4ms/step - loss: 0.0106 -
val_loss:
0.0104
Epoch 53/200
108/108 [==============================] - 1s 5ms/step - loss: 0.0106 -
val_loss:
0.0104
Epoch 54/200
108/108 [==============================] - 1s 6ms/step - loss: 0.0105 -
val_loss:
0.0115
Epoch 55/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0109
Epoch 56/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0113
Epoch 57/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0112
Epoch 58/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0117
Epoch 59/200
108/108 [==============================] - 0s 4ms/step - loss: 0.0105 -
val_loss:
0.0119
Epoch 60/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0106 -
val_loss:
0.0107
Epoch 61/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0104
Epoch 62/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0105
Epoch 63/200
108/108 [==============================] - 1s 5ms/step - loss: 0.0105 -
val_loss:
0.0105
Epoch 64/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0110
Epoch 65/200
```

```
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0124
Epoch 66/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0122
Epoch 67/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0106
Epoch 68/200
108/108 [==============================] - 0s 4ms/step - loss: 0.0105 -
val_loss:
0.0107
Epoch 69/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0107
Epoch 70/200
108/108 [==============================] - 1s 5ms/step - loss: 0.0105 -
val_loss:
0.0109
Epoch 71/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0103
Epoch 72/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0104
Epoch 73/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0104
Epoch 74/200
108/108 [==============================] - 0s 4ms/step - loss: 0.0105 -
val_loss:
0.0106
Epoch 75/200
108/108 [==============================] - 0s 4ms/step - loss: 0.0105 -
val_loss:
0.0118
Epoch 76/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0104
Epoch 77/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0112
Epoch 78/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0109
Epoch 79/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0105
Epoch 80/200
```

```
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0103
Epoch 81/200
108/108 [==============================] - 1s 6ms/step - loss: 0.0105 -
val_loss:
0.0104
Epoch 82/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0109
Epoch 83/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0104
Epoch 84/200
108/108 [==============================] - 0s 4ms/step - loss: 0.0105 -
val_loss:
0.0112
Epoch 85/200
108/108 [==============================] - 0s 4ms/step - loss: 0.0105 -
val_loss:
0.0108
Epoch 86/200
108/108 [==============================] - 0s 4ms/step - loss: 0.0105 -
val_loss:
0.0107
Epoch 87/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0103
Epoch 88/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0109
Epoch 89/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0104
Epoch 90/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0108
Epoch 91/200
108/108 [==============================] - 1s 5ms/step - loss: 0.0105 -
val_loss:
0.0106
Epoch 92/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0108
Epoch 93/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0108
Epoch 94/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0105
Epoch 95/200
```

```
108/108 [==============================] - 0s 3ms/step - loss: 0.0104 -
val_loss:
0.0106
Epoch 96/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0104
Epoch 97/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0104
Epoch 98/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0110
Epoch 99/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0114
Epoch 100/200
108/108 [==============================] - 0s 4ms/step - loss: 0.0105 -
val_loss:
0.0104
Epoch 101/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0104
Epoch 102/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0103
Epoch 103/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0104
Epoch 104/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0104 -
val_loss:
0.0104
Epoch 105/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0104
Epoch 106/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0104
Epoch 107/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0104
Epoch 108/200
108/108 [==============================] - 0s 4ms/step - loss: 0.0105 -
val_loss:
0.0107
Epoch 109/200
108/108 [==============================] - 0s 4ms/step - loss: 0.0105 -
val_loss:
0.0103
Epoch 110/200
```

```
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0104
Epoch 111/200
108/108 [==============================] - 0s 4ms/step - loss: 0.0105 -
val_loss:
0.0107
Epoch 112/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0104
Epoch 113/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0103
Epoch 114/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0104
Epoch 115/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0105
Epoch 116/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0104 -
val_loss:
0.0105
Epoch 117/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0103
Epoch 118/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0104 -
val_loss:
0.0113
Epoch 119/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0108
Epoch 120/200
108/108 [==============================] - 1s 5ms/step - loss: 0.0105 -
val_loss:
0.0105
Epoch 121/200
108/108 [==============================] - 0s 4ms/step - loss: 0.0105 -
val_loss:
0.0107
Epoch 122/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0103
Epoch 123/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0116
Epoch 124/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0111
Epoch 125/200
```

```
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0103
Epoch 126/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0104
Epoch 127/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0103
Epoch 128/200
108/108 [==============================] - 1s 5ms/step - loss: 0.0105 -
val_loss:
0.0105
Epoch 129/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0106
Epoch 130/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0104
Epoch 131/200
108/108 [==============================] - 0s 4ms/step - loss: 0.0105 -
val_loss:
0.0105
Epoch 132/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0104
Epoch 133/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0107
Epoch 134/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0109
Epoch 135/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0104
Epoch 136/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0104
Epoch 137/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0104
Epoch 138/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0109
Epoch 139/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0114
Epoch 140/200
```

```
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0103
Epoch 141/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0109
Epoch 142/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0108
Epoch 143/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0104
Epoch 144/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0106
Epoch 145/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0103
Epoch 146/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0113
Epoch 147/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0103
Epoch 148/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0106
Epoch 149/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0107
Epoch 150/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0110
Epoch 151/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0107
Epoch 152/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0103
Epoch 153/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0103
Epoch 154/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0105
Epoch 155/200
```

```
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0104
Epoch 156/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0106
Epoch 157/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0106
Epoch 158/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0104
Epoch 159/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0104
Epoch 160/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0104
Epoch 161/200
108/108 [==============================] - 1s 5ms/step - loss: 0.0105 -
val_loss:
0.0106
Epoch 162/200
108/108 [==============================] - 0s 4ms/step - loss: 0.0105 -
val_loss:
0.0104
Epoch 163/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0104
Epoch 164/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0107
Epoch 165/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0104 -
val_loss:
0.0107
Epoch 166/200
108/108 [==============================] - 0s 4ms/step - loss: 0.0105 -
val_loss:
0.0107
Epoch 167/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0103
Epoch 168/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0104
Epoch 169/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0106
Epoch 170/200
```

```
108/108 [==============================] - 1s 5ms/step - loss: 0.0105 -
val_loss:
0.0103
Epoch 171/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0104 -
val_loss:
0.0104
Epoch 172/200
108/108 [==============================] - 0s 4ms/step - loss: 0.0105 -
val_loss:
0.0109
Epoch 173/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0113
Epoch 174/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0108
Epoch 175/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0104
Epoch 176/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0126
Epoch 177/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0104
Epoch 178/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0104
Epoch 179/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0104 -
val_loss:
0.0113
Epoch 180/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0104
Epoch 181/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0103
Epoch 182/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0105
Epoch 183/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0104
Epoch 184/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0105
Epoch 185/200
```

```
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0105
Epoch 186/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0104
Epoch 187/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0104
Epoch 188/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0104 -
val_loss:
0.0110 Epoch
189/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0111
Epoch 190/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0107
Epoch 191/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0112
Epoch 192/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0104 Epoch
193/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0111
Epoch 194/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0104
Epoch 195/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0111
Epoch 196/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0103 Epoch
197/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0105
Epoch 198/200
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0119
Epoch 199/200
108/108 [==============================] - 1s 5ms/step - loss: 0.0105 -
val_loss:
0.0104
Epoch 200/200
```

```
108/108 [==============================] - 0s 3ms/step - loss: 0.0105 -
val_loss:
0.0105
```

9. Визуализируйте набор данных в виде точечного графика и прогноз нейронной сети в виде поверхности в трехмерном пространстве.

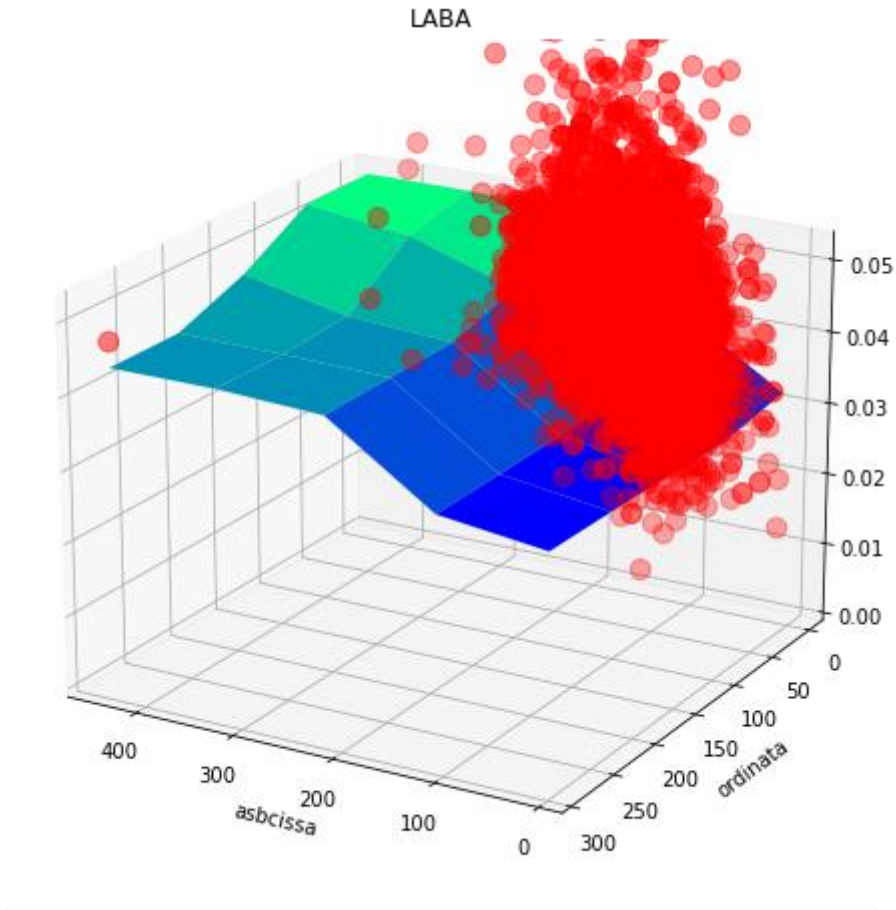
In [11]:
```
xs = data2.values[:,0] ys =
 data2.values[:,1] zs =
 data1['features/chlorides']
 n_plot = 5
```

In [12]:
```
x_plot = np.linspace(np.min(xs), np.max(xs), n_plot)
y_plot = np.linspace(np.min(ys), np.max(ys), n_plot)
x_mesh, y_mesh = np.meshgrid(x_plot, y_plot)
```

In [13]:
```
x_plot2 = np.reshape(x_mesh, [n_plot**2,1])
y_plot2 = np.reshape(y_mesh, [n_plot**2,1])
xy_2 = np.hstack([x_plot2, y_plot2])
```

In [14]:
```
z = large_model.predict(xy_2)
z_mesh = z.reshape((n_plot,
n_plot))
```

In [19]:
```
from matplotlib import cm fig =
plt.figure(figsize=(10, 8)) ax =
fig.add_subplot(111, projection='3d') surf =
ax.plot_surface(x_mesh, y_mesh, z_mesh, \
       rstride=1, cstride=1, linewidth=0.05, cmap=cm.winter, antialiased=True,
\        edgecolors='gray')
ax.scatter( xs, ys, zs, s=100, c='r' )
plt.title('LABA')
plt.xlabel('asbcissa')
plt.ylabel('ordinata')
ax.set_zlabel('Z', fontsize=14)
ax.set_zlim(0., z_mesh.max())
ax.view_init(elev = 20, azim = 120)
```



10. Разбейте набор данных из двух признаков и отклика на обучающую и тестовую выборки и постройте кривые обучения для заданного показателя качества в зависимости от количества точек в обучающей выборке, подписывая оси и рисунок и создавая легенду.

In [16]:

```
def train_test_split(X, y, test_ratio=0.2, seed=None):
"""возвращает X_train, X_test, y_train, y_test"""
assert X.shape[0] == y.shape[0], \
        "Размер X должен быть равен размеру y"
assert 0.0 <= test_ratio <= 1.0, \
"Неверное значение test_ratio"
     if
seed:
        np.random.seed(seed)

    shuffled_indexes = np.random.permutation(len(X))

    test_size = int(len(X) * test_ratio)
test_indexes = shuffled_indexes[:test_size]
train_indexes = shuffled_indexes[test_size:]
    X_train = X[train_indexes]
y_train = y[train_indexes]

    X_test = X[test_indexes]
y_test = y[test_indexes]

    return X_train, X_test, y_train, y_test
X_train, X_test, y_train, y_test = train_test_split(data2.values,
data1['features X_train.shape
```

Out[16]:

```
(3919, 2)
```

In [17]:

```
train_score = [] test_score = [] for i
in range(1, 3919, int(3919/30)):
large_model = tf.keras.Sequential([
feature_normalizer,
        tf.keras.layers.Dense(units=64, activation='relu'),
tf.keras.layers.Dense(units=1)
    ])
    large_model.compile(loss='mse')
    large_model.fit(X_train[:i], y_train[:i], epochs=10, verbose=0)

    y_train_predict = large_model.predict(X_train[:i])
    train_score.append(mean_absolute_error(y_train[:i], y_train_predict))

    y_test_predict = large_model.predict(X_test)
test_score.append(mean_absolute_error(y_test, y_test_predict))     print('--
>', i, ' done')
```

```
 --> 1   done
--> 131   done --
  > 261   done
 --> 391   done
 --> 521   done
--> 651   done --
  > 781   done
 --> 911   done
 --> 1041   done
--> 1171   done --
  > 1301   done
 --> 1431   done
 --> 1561   done
```

```
--> 1691  done --
  > 1821  done
 --> 1951  done
 --> 2081  done
--> 2211  done --
  > 2341  done
 --> 2471  done
 --> 2601  done
--> 2731  done --
  > 2861  done
 --> 2991  done
 --> 3121  done
--> 3251  done --
  > 3381  done
 --> 3511  done
 --> 3641  done
 --> 3771  done
 --> 3901  done
```

In [18]:

```
plt.title('LABA')
plt.xlabel('asbcissa')
plt.ylabel('ordinata')
plt.plot([i for i in range(1, len(X_train), int(3919/30))], train_score,
label="tr plt.plot([i for i in range(1, len(X_train), int(3919/30))],
test_score, label="tes plt.legend();
```