

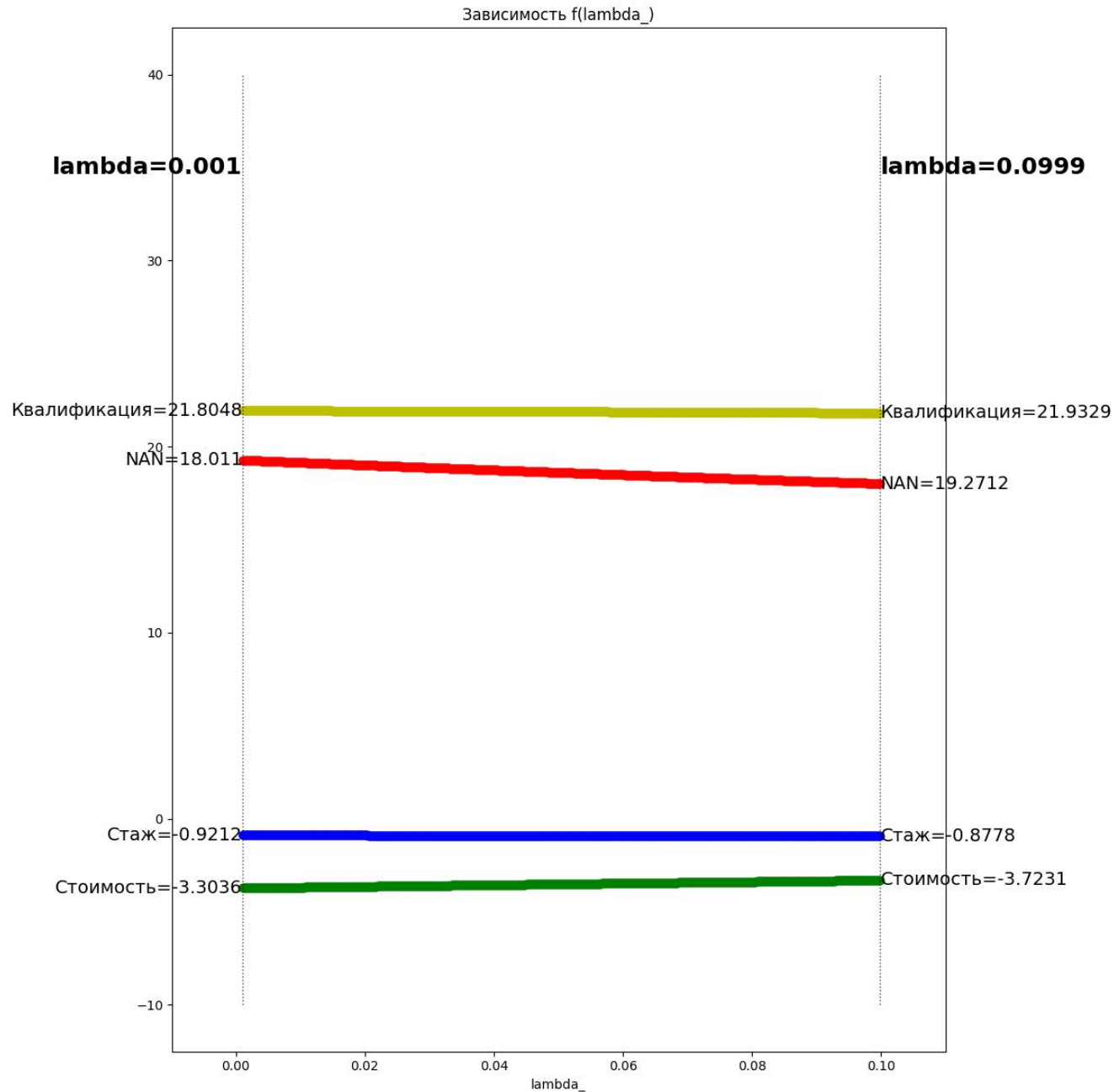
```
1 # 1. Постройте график зависимости весов всех признаков от  $\lambda$  в L2-регуляризации
   (на данных из урока).
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 X = np.array([[1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
6               [1, 1, 2, 1, 3, 0, 5, 10, 1, 2], # стаж
7               [500, 700, 750, 600, 1450, # средняя стоимость занятия
8               800, 1500, 2000, 450, 1000],
9               [1, 1, 2, 1, 2, 1, 3, 3, 1, 2]]) # квалификация репетитора
10
11 y = [45, 55, 50, 59, 65, 35, 75, 80, 50, 60]
12
13 w = np.dot(
14     np.dot(
15         np.linalg.inv(
16             np.dot(X, X.T)
17         ),
18         X),
19     y)
20 X_norm = X.copy()
21 X_norm = X_norm.astype(np.float64)
22 X_norm[1] = (X[1] - X[1].min()) / (X[1].max() - X[1].min())
23 X_norm[2] = (X[2] - X[2].min()) / (X[2].max() - X[2].min())
24
25 X_st = X.copy().astype(np.float64)
26 X_st[1] = (X[1] - X[1].mean()) / X[1].std()
27 X_st[2] = (X[2] - X[2].mean()) / X[2].std()
```

```
28
29
30 def calc_mse(y, y_pred):
31     err = np.mean((y - y_pred) ** 2)
32     return err
33
34
35 def eval_model_reg2(X, y, iterations, alpha=1e-4, lambda_=1e-8):
36     np.random.seed(42)
37     W = np.random.randn(X.shape[0])
38     n = X.shape[1]
39     print(f'lambda={lambda_}')
40     for i in range(1, iterations + 1):
41         y_pred = np.dot(W, X)
42         err = calc_mse(y, y_pred)
43         W -= alpha * (1 / n * 2 * np.dot((y_pred - y), X.T) + 2 * lambda_ * W)
44         if i % (iterations / 10) == 0:
45             print(i, W, err)
46     return W
47
48
49 level = 0.1
50 delta = 0.0001
51 lambda_ = 0.001
52
53 fig, axes = plt.subplots(1, 1)
54 fig.set_figwidth(12)
55 fig.set_figheight(12)
```

```
56
57 first = False
58
59 while lambda_ < level:
60     W = eval_model_reg2(X_st, y, iterations=800, alpha=0.003, lambda_=lambda_)
61     if not first:
62         data = np.array([[lambda_, W[0], W[1], W[2], W[3]], ])
63         first = True
64     else:
65         data = np.append(data, [[lambda_, W[0], W[1], W[2], W[3]]], axis=0)
66
67     lambda_ += delta
68
69 _data = data[:, 1:5]
70
71 color = np.sort(np.random.random(len(data)))
72 axes.title.set_text('Зависимость f(lambda_)', )
73
74 axes.set_xlabel('lambda_')
75
76 axes.set_xlim((0.0001 - 0.01, level + 0.01))
77
78 # Vertical Lines
79 axes.vlines(x=data[:, 0].min(), ymin=-10, ymax=40, color='black', alpha=0.7,
80            linewidth=1, linestyle='dotted')
81 axes.vlines(x=data[:, 0].max(), ymin=-10, ymax=40, color='black', alpha=0.7,
82            linewidth=1, linestyle='dotted')
81
```

```
82 axes.text(data[:, 0].min(), 35, f'lambda={round(data[:, 0].min(), 5)}',
    horizontalalignment='right',
83         verticalalignment='center', fontdict={'size': 18, 'weight': 700})
84 axes.text(data[:, 0].max(), 35, f'lambda={round(data[:, 0].max(), 5)}',
    horizontalalignment='left',
85         verticalalignment='center', fontdict={'size': 18, 'weight': 700})
86
87 axes.scatter(data[:, 0], _data[:, 0], c='r')
88 axes.text(data[:, 0].min(), _data[0:1, 0], f'NAN={round(float(_data[-1, 0]), 4)}',
89         horizontalalignment='right', verticalalignment='center', fontdict={'size'
    : 14})
90 axes.text(data[:, 0].max(), _data[-1, 0], f'NAN={round(float(_data[0:1, 0]), 4)}',
91         horizontalalignment='left', verticalalignment='center', fontdict={'size'
    : 14})
92
93 axes.scatter(data[:, 0], _data[:, 1], c='b')
94 axes.text(data[:, 0].min(), _data[0:1, 1], f'Стаж={round(float(_data[-1, 1]), 4)}',
95         horizontalalignment='right', verticalalignment='center', fontdict={'size'
    : 14})
96 axes.text(data[:, 0].max(), _data[-1, 1], f'Стаж={round(float(_data[0:1, 1]), 4)}',
97         horizontalalignment='left', verticalalignment='center', fontdict={'size'
    : 14})
98
99 axes.scatter(data[:, 0], _data[:, 2], c='g')
100 axes.text(data[:, 0].min(), _data[0:1, 2], f'Стоимость={round(float(_data[-1, 2]),
    4)}',
101         horizontalalignment='right', verticalalignment='center', fontdict={'size'
    : 14})
```

```
102 axes.text(data[:, 0].max(), _data[-1, 2], f'Стоимость={round(float(_data[0:1, 2]),
    4)}',
103             horizontalalignment='left', verticalalignment='center', fontdict={'size'
    : 14})
104
105
106 axes.scatter(data[:, 0], _data[:, 3], c='y')
107 axes.text(data[:, 0].min(), _data[0:1, 3], f'Квалификация={round(float(_data[-1, 3]
    ), 4)}',
108             horizontalalignment='right', verticalalignment='center', fontdict={'size'
    : 14})
109 axes.text(data[:, 0].max(), _data[-1, 3], f'Квалификация={round(float(_data[0:1, 3]
    ), 4)}',
110             horizontalalignment='left', verticalalignment='center', fontdict={'size'
    : 14})
111
112
113 plt.show()
114 # Ответ на периоде  $\lambda$  от 0.001 до 0.01 зависимость фич меняется слабо.
115
```



116 # 2. Можно ли к одному и тому же признаку применить сразу и нормализацию, и стандартизацию?

117 # Ответ: Можно, только нужно обязательно тогда нормализовать/стандартизировать все колонки данных.