Задание 1.
Исходные данные:
Сгенерировать датасет при помощи sklearn.datasets.make_regression и обучить линейную модель при помощи градиентного и стохастического градиентного спуска. Нанести среднеквадратичную ошибку для обоих методов на один график, сделать выводы о разнице скорости сходимости каждого из методов.
Решение:

```
Python 3.8.10 (default, Sep 28 2021, 16:10:42)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> inport numpy as np
>>> import pandas as pd
>>> import scipy
>>> import sklearn
>>> import matplotlib.pyplot as plt
>>> from sklearn import linear_model
>>> from sklearn.datasets import make_regression
>>> from sklearn.preprocessing import StandardScaler
>>> X, y = make_regression(n_samples=100, n_features=2, n_informative=1)
>>> X = StandardScaler().fit_transform(X)
>>> y = StandardScaler().fit_transform(y.reshape(len(y),1))[:,0]
>>> X.T
array([[ 0.29641609,  0.18435555,  1.56282981, -1.87174117, -1.35819696,
        -1.28154879, -1.53017307,  0.60171134, -1.35369398,  0.71324208,
         2.2924239 , -1.31517703, -1.80279397,  0.28114165, -0.42418098,
         0.73960835,  1.30379815,  1.00618135,  0.87085527, -0.65770353,
        -1.0996938 ,  0.89266656, -0.11044574,  0.14446159,  1.00330804,
        -0.09258082, -1.44402779,  0.45523787,  0.83949204,  0.58347771,
        -0.34364136, -1.59625715, -0.33755532, -0.14819994,  1.19116667,
         0.03673636,  0.70661645,  1.45005296, -2.00169995, -1.01142351,
         2.09531528, -0.23986743, -0.62199351, -0.020663  ,  0.31702925,
         0.47574612,  0.29661622,  0.40752228,  1.01258092,  0.01508083,
        -0.19098135,  0.59509783,  0.01983254, -0.61702834, -0.06127921,
        -0.09454695,  0.04442116, -0.32209776,  0.85715876,  0.01074777,
         0.40326984, -1.71393396, -1.59454631,  0.20855093,  0.59561353,
        -1.13618919, -0.35983719,  2.63543757, -0.10239213,  1.36178632,
         0.69792018, -0.18378524,  0.84354646, -0.11485754,  0.12678892,
        -2.10737702, -0.43093649, -0.90852783,  0.15734307, -0.82812196,
         1.4945563 ,  1.06674013, -0.22033628,  0.09009077, -1.26278465,
         1.80148432,  0.86905094, -1.52389506,  0.10079632,  0.63290663,
        -0.20285194,  2.07770682, -0.84354664, -0.16712202,  0.32324133,
         0.10225882, -1.10152313, -1.06310247, -1.06520079, -0.01195773],
       [ 0.42999142,  1.40653267, -2.42049633, -0.18837073,  0.50686163,
        -0.14585492, -0.55379512, -0.77331413,  1.4327685 , -1.11452436,
        -0.87785018, -0.19100794,  0.92402773,  0.6701648 ,  0.48475221,
         0.48019739, -0.18540193,  0.20788989, -1.44815315,  0.21199312,
        -0.36102028,  0.82566801, -0.32765262, -2.08216147,  0.1870284 ,
        -1.48996494,  0.37175247, -0.49898324, -0.02706825, -0.55152335,
        -1.1930628 ,  0.10561035,  0.66528746,  0.91606657, -0.04271604,
         0.17017245,  0.97593894,  0.79759916,  0.57583918,  0.40303656,
         0.68038043,  2.08465073,  1.70550271,  1.16674327,  2.08725797,
         0.25007398, -0.5894363 , -2.12844327,  0.20095669,  1.42088647,
```

```
        1.04029667,  0.18192649,  0.58416187,  0.12616218, -0.37778432,
        0.11570221,  0.15191418, -0.89230816, -1.59740389, -1.36614956,
       -0.23100084,  0.50232838, -0.48129709,  0.01224895, -0.20872863,
       -0.91996068,  0.56903168,  1.06672029, -0.65449258, -0.01312838,
        0.58109226, -1.91786949,  0.06940413, -0.76239204, -0.4910064 ,
        0.50698876, -2.47851518,  0.82236501, -0.43649981, -0.92140087,
        0.63587015,  0.72833762, -1.69249006,  0.07819722,  0.64842634,
       -0.76071919,  1.37024613, -0.55913053, -1.52996584,  0.83300059,
       -0.34285122,  2.70704036,  0.10286685,  0.28597572, -0.22458342,
       -1.39930906,  1.67775001,  1.11371884, -1.55186815,  0.14425268]])
>>> y_pred1 = 35 * np.ones(100) + X.T[1]*5
>>> y_pred2 = 40 * np.ones(100) + X.T[1]*7.5
>>> y_pred1, y_pred2
(array([37.14995711, 42.03266336, 22.89751837, 34.05814635, 37.53430817,
       34.27072542, 32.2310244 , 31.13342933, 42.1638425 , 29.42737819,
       30.61074908, 34.04496032, 39.62013863, 38.35082402, 37.42376106,
       37.40098697, 34.07299036, 36.03944945, 27.75923425, 36.05996559,
       33.19489862, 39.12834006, 33.36173692, 24.58919263, 35.935142  ,
       27.55017529, 36.85876237, 32.50508378, 34.86465875, 32.24238325,
       29.03468602, 35.52805176, 38.32643728, 39.58033284, 34.78641978,
       35.85086224, 39.87969468, 38.98799582, 37.87919588, 37.01518278,
       38.40190217, 45.42325365, 43.52751357, 40.83371635, 45.43628986,
       36.25036992, 32.05281851, 24.35778365, 36.00478343, 42.10443234,
       40.20148335, 35.90963244, 37.92080933, 35.63081088, 33.11107839,
       35.57851105, 35.75957092, 30.53845919, 27.01298055, 28.16925222,
       33.84499581, 37.51164192, 32.59351455, 35.06124474, 33.95635684,
       30.40019659, 37.84515839, 40.33360144, 31.72753709, 34.93435811,
       37.90546128, 25.41065256, 35.34702065, 31.1880398 , 32.544968  ,
       37.53494379, 22.6074241 , 39.11182507, 32.81750093, 30.39299567,
       38.17935073, 38.64168812, 26.53754971, 35.39098611, 38.24213168,
       31.19640406, 41.85123064, 32.20434737, 27.3501708 , 39.16500297,
       33.2857439 , 48.53520179, 35.51433426, 36.42987862, 33.87708291,
       28.00345468, 43.38875005, 40.5685942 , 27.24065923, 35.7212634 ]), array([43.22493567,
50.54899505, 21.84627756, 38.58721953, 43.80146225,
       38.90608814, 35.8465366 , 34.20014399, 50.74576375, 31.64106728,
       33.41612362, 38.56744048, 46.93020795, 45.02623604, 43.63564159,
       43.60148045, 38.60948554, 41.55917418, 29.13885138, 41.58994839,
       37.29234792, 46.19251008, 37.54260537, 24.38378895, 41.40271299,
       28.82526294, 42.78814355, 36.25762566, 39.79698812, 35.86357488,
       31.05202902, 40.79207764, 44.98965592, 46.87049925, 39.67962967,
       41.27629335, 47.31954203, 45.98199373, 44.31879382, 43.02277417,
       45.10285325, 55.63488047, 52.79127036, 48.75057452, 55.65443479,
       41.87555489, 35.57922776, 24.03667547, 41.50717514, 50.65664852,
       47.80222503, 41.36444867, 44.38121399, 40.94621632, 37.16661759,
       40.86776657, 41.13935637, 33.30768878, 28.01947082, 29.75387833,
       38.26749372, 43.76746289, 36.39027182, 40.09186711, 38.43453525,
       33.10029489, 44.26773758, 48.00040217, 35.09130564, 39.90153716,
       44.35819192, 25.61597884, 40.52053098, 34.2820597 , 36.31745199,
       43.80241569, 21.41113615, 46.16773761, 36.72625139, 33.08949351,
       44.7690261 , 45.46253217, 27.30632457, 40.58647917, 44.86319752,
       34.29460609, 50.27684595, 35.80652106, 28.5252562 , 46.24750446,
       37.42861584, 60.30280269, 40.77150139, 42.14481793, 38.31562436,
```

```
       29.50518202, 52.58312508, 48.3528913 , 28.36098884, 41.0818951 ]))
>>> err1 = np.sum(y - y_pred1)
>>> err2 = np.sum(y - y_pred2)
>>> err1, err2
(-3500.0, -4000.0000000000005)
>>> mae_1 = np.sum(np.abs(y - y_pred1)) / 100
>>> mae_2 = np.sum(np.abs(y - y_pred2)) / 100
>>> mae_1, mae_2
(35.0, 40.00000000000001)
>>> mse_1 = np.mean((y - y_pred1) ** 2)
>>> mse_2 = np.mean((y - y_pred2) ** 2)
>>> mse_1, mse_2
(1241.0, 1642.2500000000002)
>>> X
array([[ 0.29641609,  0.42999142],
       [ 0.18435555,  1.40653267],
       [ 1.56282981, -2.42049633],
       [-1.87174117, -0.18837073],
       [-1.35819696,  0.50686163],
       [-1.28154879, -0.14585492],
       [-1.53017307, -0.55379512],
       [ 0.60171134, -0.77331413],
       [-1.35369398,  1.4327685 ],
       [ 0.71324208, -1.11452436],
       [ 2.2924239 , -0.87785018],
       [-1.31517703, -0.19100794],
       [-1.80279397,  0.92402773],
       [ 0.28114165,  0.6701648 ],
       [-0.42418098,  0.48475221],
       [ 0.73960835,  0.48019739],
       [ 1.30379815, -0.18540193],
       [ 1.00618135,  0.20788989],
       [ 0.87085527, -1.44815315],
       [-0.65770353,  0.21199312],
       [-1.0996938 , -0.36102028],
       [ 0.89266656,  0.82566801],
       [-0.11044574, -0.32765262],
       [ 0.14446159, -2.08216147],
       [ 1.00330804,  0.1870284 ],
       [-0.09258082, -1.48996494],
       [-1.44402779,  0.37175247],
       [ 0.45523787, -0.49898324],
       [ 0.83949204, -0.02706825],
       [ 0.58347771, -0.55152335],
       [-0.34364136, -1.1930628 ],
       [-1.59625715,  0.10561035],
       [-0.33755532,  0.66528746],
       [-0.14819994,  0.91606657],
       [ 1.19116667, -0.04271604],
       [ 0.03673636,  0.17017245],
       [ 0.70661645,  0.97593894],
       [ 1.45005296,  0.79759916],
```

```
[-2.00169995,  0.57583918],
[-1.01142351,  0.40303656],
[ 2.09531528,  0.68038043],
[-0.23986743,  2.08465073],
[-0.62199351,  1.70550271],
[-0.020663  ,  1.16674327],
[ 0.31702925,  2.08725797],
[ 0.47574612,  0.25007398],
[ 0.29661622, -0.5894363 ],
[ 0.40752228, -2.12844327],
[ 1.01258092,  0.20095669],
[ 0.01508083,  1.42088647],
[-0.19098135,  1.04029667],
[ 0.59509783,  0.18192649],
[ 0.01983254,  0.58416187],
[-0.61702834,  0.12616218],
[-0.06127921, -0.37778432],
[-0.09454695,  0.11570221],
[ 0.04442116,  0.15191418],
[-0.32209776, -0.89230816],
[ 0.85715876, -1.59740389],
[ 0.01074777, -1.36614956],
[ 0.40326984, -0.23100084],
[-1.71393396,  0.50232838],
[-1.59454631, -0.48129709],
[ 0.20855093,  0.01224895],
[ 0.59561353, -0.20872863],
[-1.13618919, -0.91996068],
[-0.35983719,  0.56903168],
[ 2.63543757,  1.06672029],
[-0.10239213, -0.65449258],
[ 1.36178632, -0.01312838],
[ 0.69792018,  0.58109226],
[-0.18378524, -1.91786949],
[ 0.84354646,  0.06940413],
[-0.11485754, -0.76239204],
[ 0.12678892, -0.4910064 ],
[-2.10737702,  0.50698876],
[-0.43093649, -2.47851518],
[-0.90852783,  0.82236501],
[ 0.15734307, -0.43649981],
[-0.82812196, -0.92140087],
[ 1.4945563 ,  0.63587015],
[ 1.06674013,  0.72833762],
[-0.22033628, -1.69249006],
[ 0.09009077,  0.07819722],
[-1.26278465,  0.64842634],
[ 1.80148432, -0.76071919],
[ 0.86905094,  1.37024613],
[-1.52389506, -0.55913053],
[ 0.10079632, -1.52996584],
[ 0.63290663,  0.83300059],
```

```
       [-0.20285194, -0.34285122],
       [ 2.07770682,  2.70704036],
       [-0.84354664,  0.10286685],
       [-0.16712202,  0.28597572],
       [ 0.32324133, -0.22458342],
       [ 0.10225882, -1.39930906],
       [-1.10152313,  1.67775001],
       [-1.06310247,  1.11371884],
       [-1.06520079, -1.55186815],
       [-0.01195773,  0.14425268]])
>>> X.shape
(100, 2)
>>> X.T
array([[ 0.29641609,  0.18435555,  1.56282981, -1.87174117, -1.35819696,
        -1.28154879, -1.53017307,  0.60171134, -1.35369398,  0.71324208,
         2.2924239 , -1.31517703, -1.80279397,  0.28114165, -0.42418098,
         0.73960835,  1.30379815,  1.00618135,  0.87085527, -0.65770353,
        -1.0996938 ,  0.89266656, -0.11044574,  0.14446159,  1.00330804,
        -0.09258082, -1.44402779,  0.45523787,  0.83949204,  0.58347771,
        -0.34364136, -1.59625715, -0.33755532, -0.14819994,  1.19116667,
         0.03673636,  0.70661645,  1.45005296, -2.00169995, -1.01142351,
         2.09531528, -0.23986743, -0.62199351, -0.020663  ,  0.31702925,
         0.47574612,  0.29661622,  0.40752228,  1.01258092,  0.01508083,
        -0.19098135,  0.59509783,  0.01983254, -0.61702834, -0.06127921,
        -0.09454695,  0.04442116, -0.32209776,  0.85715876,  0.01074777,
         0.40326984, -1.71393396, -1.59454631,  0.20855093,  0.59561353,
        -1.13618919, -0.35983719,  2.63543757, -0.10239213,  1.36178632,
         0.69792018, -0.18378524,  0.84354646, -0.11485754,  0.12678892,
        -2.10737702, -0.43093649, -0.90852783,  0.15734307, -0.82812196,
         1.4945563 ,  1.06674013, -0.22033628,  0.09009077, -1.26278465,
         1.80148432,  0.86905094, -1.52389506,  0.10079632,  0.63290663,
        -0.20285194,  2.07770682, -0.84354664, -0.16712202,  0.32324133,
         0.10225882, -1.10152313, -1.06310247, -1.06520079, -0.01195773],
       [ 0.42999142,  1.40653267, -2.42049633, -0.18837073,  0.50686163,
        -0.14585492, -0.55379512, -0.77331413,  1.4327685 , -1.11452436,
        -0.87785018, -0.19100794,  0.92402773,  0.6701648 ,  0.48475221,
         0.48019739, -0.18540193,  0.20788989, -1.44815315,  0.21199312,
        -0.36102028,  0.82566801, -0.32765262, -2.08216147,  0.1870284 ,
        -1.48996494,  0.37175247, -0.49898324, -0.02706825, -0.55152335,
        -1.1930628 ,  0.10561035,  0.66528746,  0.91606657, -0.04271604,
         0.17017245,  0.97593894,  0.79759916,  0.57583918,  0.40303656,
         0.68038043,  2.08465073,  1.70550271,  1.16674327,  2.08725797,
         0.25007398, -0.5894363 , -2.12844327,  0.20095669,  1.42088647,
         1.04029667,  0.18192649,  0.58416187,  0.12616218, -0.37778432,
         0.11570221,  0.15191418, -0.89230816, -1.59740389, -1.36614956,
        -0.23100084,  0.50232838, -0.48129709,  0.01224895, -0.20872863,
        -0.91996068,  0.56903168,  1.06672029, -0.65449258, -0.01312838,
         0.58109226, -1.91786949,  0.06940413, -0.76239204, -0.4910064 ,
         0.50698876, -2.47851518,  0.82236501, -0.43649981, -0.92140087,
         0.63587015,  0.72833762, -1.69249006,  0.07819722,  0.64842634,
        -0.76071919,  1.37024613, -0.55913053, -1.52996584,  0.83300059,
        -0.34285122,  2.70704036,  0.10286685,  0.28597572, -0.22458342,
```

```
        -1.39930906,  1.67775001,  1.11371884, -1.55186815,  0.14425268]])
>>> X.T.shape
(2, 100)
>>> all(X.T @ y == np.dot(X.T, y))
True
>>> W = np.linalg.inv(np.dot(X.T, X)) @X.T @ y
>>> W
array([1.38777878e-17, 1.00000000e+00])
>>> y_pred3 = W[0] * X[0] + W[1] * X[1]
>>> y_pred3
array([0.18435555, 1.40653267])
>>> def calc_mae(y, y_pred):
...     err = np.mean(np.abs(y - y_pred))
...     return err
...
>>> def calc_mse(y, y_pred):
...     err = np.mean((y- y_pred) ** 2) # <=> 1/n * np.sum((y_pred - y) ** 2)
...     return err
...
>>> calc_mae(y, y_pred1), calc_mse(y, y_pred1)
(35.0, 1241.0)
>>> calc_mae(y, y_pred2), calc_mse(y, y_pred2)
(40.00000000000001, 1642.2500000000002)
>>> plt.plot(X, y)
[<matplotlib.lines.Line2D object at 0x7f696ccbcac0>, <matplotlib.lines.Line2D object at 0x7f696ccbcc10>]
>>> plt.show()
>>> W = np.array([1, 0.5])
>>> W
array([1. , 0.5])
>>> gradient_form_direct = 1/100 * 2 * np.sum(X[0] * W[0] - y[0])
>>> print(gradient_form_direct)
-0.0026715066107574326
>>> print(W[0] - gradient_form_direct)
1.0026715066107574
>>>
```

Проанализировав полученные данные: считаю, что стахостический градиентный спуск, отработал более быстро и у него более лучшая сходимость, чем у обычного градиентного спуска.