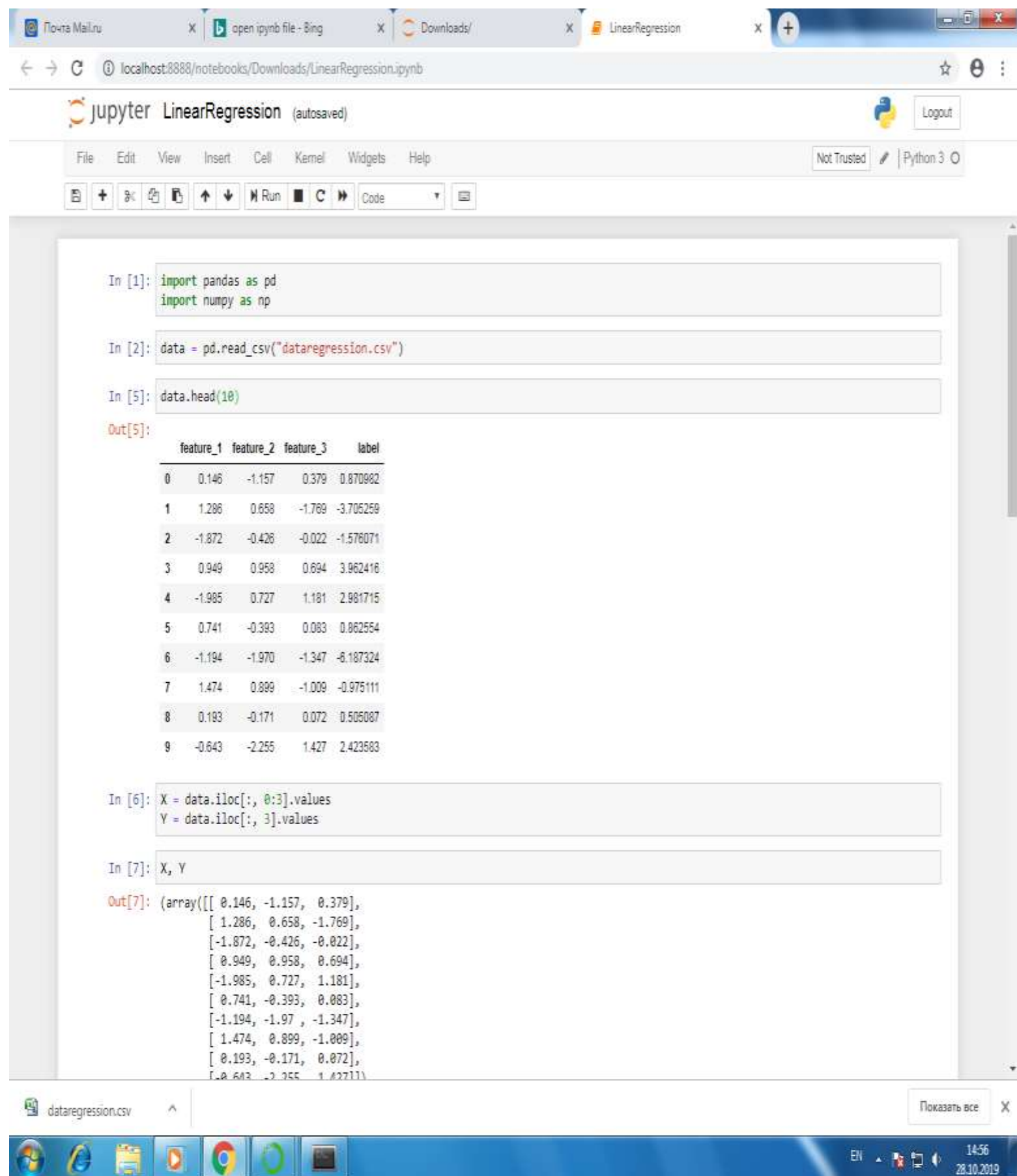


Лабораторное занятие 6. Реализация метода линейной регрессии в Python.

6-зертханалық сабақ. Python да сызықты регрессия әдісін жүзеге асыру.

Написать следующий код



The screenshot shows a Jupyter Notebook interface with the following content:

```
In [1]: import pandas as pd
import numpy as np

In [2]: data = pd.read_csv("dataregression.csv")

In [5]: data.head(10)
```

Out[5]:

	feature_1	feature_2	feature_3	label
0	0.146	-1.157	0.379	0.870982
1	1.286	0.658	-1.769	-3.705259
2	-1.872	-0.426	-0.022	-1.576071
3	0.949	0.958	0.694	3.962416
4	-1.985	0.727	1.181	2.981715
5	0.741	-0.393	0.083	0.862554
6	-1.194	-1.970	-1.347	-6.187324
7	1.474	0.899	-1.009	-0.975111
8	0.193	-0.171	0.072	0.505087
9	-0.643	-2.255	1.427	2.423683

```
In [6]: X = data.iloc[:, 0:3].values
Y = data.iloc[:, 3].values

In [7]: X, Y

Out[7]: (array([[ 0.146, -1.157,  0.379],
 [ 1.286,  0.658, -1.769],
 [-1.872, -0.426, -0.022],
 [ 0.949,  0.958,  0.694],
 [-1.985,  0.727,  1.181],
 [ 0.741, -0.393,  0.083],
 [-1.194, -1.97 , -1.347],
 [ 1.474,  0.899, -1.009],
 [ 0.193, -0.171,  0.072],
 [-0.643, -2.255,  1.427]]),
 array([ 0.870982, -3.705259, -1.576071,  3.962416,  2.981715,  0.862554,
        -6.187324, -0.975111,  0.505087,  2.423683]))
```

The interface also shows a file explorer at the bottom with 'dataregression.csv' selected. The system tray at the bottom right indicates the date and time: 28.10.2019, 14:56.

```
In [8]: X
```

```
In [9]: X_tr = np.transpose(X)
X_tr_X = np.dot(X_tr, X)
X_tr_X_inv = np.linalg.inv(X_tr_X)
coef = np.dot(X_tr_X_inv, X_tr)
params_analytic = np.dot(coef, Y)
```

```
In [10]: params analytic
```

```
Out[10]: array([ 0.31749375,  0.78032738,  0.74979223,  3.08462768])
```

```
In [11]: teta_grad_start = np.zeros(4)
learning_rate = 0.001
epsilon = 0.00001
max_iter = 100000

def step(teta):
    return teta - learning_rate * (1.0/10) * np.dot(np.transpose(X), (np.dot(X, teta) - Y))

teta_GD = teta_grad_start
for i in range(max_iter):
    teta_GD_new = step(teta_GD)
    if np.sum(np.abs(teta_GD_new - teta_GD)) <= epsilon:
        break
    else:
        teta_GD = teta_GD_new

print(i)
```

Почта Mail.ru

open ipynb file - Bing

Downloads/

LinearRegression

+

localhost8888/notebooks/Downloads/LinearRegression.ipynb

☆ ⓘ ⋮

Jupyter LinearRegression (autosaved)

Python 3

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted

Python 3

+

%

↺

↻

↱

↲

Run

⏏

Code

⌵

⌵

```
def step(teta):
    return teta - learning_rate * (1.0/10) * np.dot(np.transpose(X), (np.dot(X, teta) - Y))

teta_GD = teta_grad_start
for i in range(max_iter):
    teta_GD_new = step(teta_GD)
    if np.sum(np.abs(teta_GD_new - teta_GD)) <= epsilon:
        break
    else:
        teta_GD = teta_GD_new

print(i)
print(teta_GD)
```

8454

[0.31673068 0.77492892 0.75208069 3.07801993]

In [12]:

from sklearn.linear_model import LinearRegression

In [13]:

LR = LinearRegression()

In [14]:

X = data.iloc[:, 0:3].values
Y = data.iloc[:, 3].values
LR.fit(X, Y)

Out[14]:

LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)

In [15]:

LR.coef_, LR.intercept_

Out[15]:

(array([0.78032738, 0.74979223, 3.08462768]), 0.31749374659996571)

In [16]:

LR.predict([0.146,-1.157,0.379])

Out[16]:

array([0.73298582])

In []:

dataregression.csv

Показать все

EN

15:00

28.10.2019