Практическое задание 6

In [1]:

```
from sympy import *
import math
import pandas as pd
init_printing()
import numpy as np
```

1. Найти частные производные первого и второго порядка. Убедиться в равенстве смешанных производных.

$$U = x^3 + 3xy^2 + z^2 - 39x - 36y + 2z + 26$$

$$U_x' = 3x^2 + 3y^2 - 39$$

$$U_{xx}'' = 6x$$

$$U_{xy}'' = 6y$$

$$U_{xz}'' = 0$$

$$U_y' = 6xy - 36$$

$$U_{yx}'' = 6y$$

$$U_{yy}'' = 6x$$

$$U_{yz}'' = 0$$

$$U_z' = 2z + 2$$

$$U_{zx}'' = 20$$

$$U''_{zy} = 0$$

$$U''_{zz} = 2$$

Видим, что смешанные производны попарно равны

Проверим

In [2]:

```
x,y,z=symbols('x y z')
f=x**3+3*x*y**2+z**2-39*x-36*y+2*z+26
f
```

Out[2]:

$$x^3 + 3xy^2 - 39x - 36y + z^2 + 2z + 26$$

In [3]:

```
for i in(x, y, z):
    print(f"Производная первого порядка по {i}: ", diff(f,i))
    for j in(x, y, z):
        print(f"Производная второго порядка по {i}{j}: ", diff(diff(f,i),j))
    print()
```

```
Производная первого порядка по x:  3*x**2 + 3*y**2 - 39
Производная второго порядка по xx:  6*x
Производная второго порядка по xy:  6*y
Производная второго порядка по xz:  0

Производная первого порядка по y:  6*x*y - 36
Производная второго порядка по yx:  6*y
Производная второго порядка по yy:  6*x
Производная второго порядка по yz:  0

Производная первого порядка по z:  2*z + 2
Производная второго порядка по zx:  0
Производная второго порядка по zy:  0
Производная второго порядка по zz:  2
```

Видим, что частные производные первого и второго порядка найдены верно

2. Найти частные производные первого и второго порядка. Убедиться в равенстве смешанных производных.

$$U = \frac{256}{x} + \frac{x^2}{y} + \frac{y^2}{z} + z^2$$

$$U'_x = -\frac{256}{x^2} + \frac{2x}{y}$$

$$U''_{xx} = \frac{512}{x^3} + \frac{2}{y}$$

$$U''_{xy} = -\frac{2x}{y^2}$$

$$U''_{xz} = 0$$

$$U'_y = -\frac{x^2}{y^2} + \frac{2y}{z}$$

$$U''_{yx} = -\frac{2x}{y^2}$$

$$U''_{yy} = \frac{2x^2}{y^3} + \frac{2}{z}$$

$$U''_{yz} = -\frac{2y}{z^2}$$

$$U'_z = -\frac{y^2}{z^2} + 2z$$

$$U''_{zx} = 0$$

$$U''_{zy} = -\frac{2y}{z^2}$$

$$U''_{zz} = \frac{2y^2}{z^3} + 2$$

Видим, что смешанные производны попарно равны

Проверим

In [4]:

```
x,y,z=symbols('x y z')
f=256/x+x**2/y+y**2/z+z**2
f
```

Out[4]:

$$\frac{x^2}{y} + \frac{y^2}{z} + z^2 + \frac{256}{x}$$

In [5]:

```
for i in(x, y, z):
    print(f"Производная первого порядка по {i}: ", diff(f,i))
    for j in(x, y, z):
        print(f"Производная второго порядка по {i}{j}: ", diff(diff(f,i),j))
    print()
```

```
Производная первого порядка по x:  2*x/y - 256/x**2
Производная второго порядка по xx:  2/y + 512/x**3
Производная второго порядка по xy:  -2*x/y**2
Производная второго порядка по xz:  0

Производная первого порядка по y:  -x**2/y**2 + 2*y/z
Производная второго порядка по yx:  -2*x/y**2
Производная второго порядка по yy:  2*x**2/y**3 + 2/z
Производная второго порядка по yz:  -2*y/z**2

Производная первого порядка по z:  -y**2/z**2 + 2*z
Производная второго порядка по zx:  0
Производная второго порядка по zy:  -2*y/z**2
Производная второго порядка по zz:  2*y**2/z**3 + 2
```

Видим, что частные производные первого и второго порядка найдены верно

3. Найти производную функции $U = x^2 + y^2 + z^2$ по направлению вектора $\vec{c}\,(-9, 8, -12)$ в точке $M(8, -12, 9)$

Для нахождения производной по направлению нам необходимо знать:

- единичный направляющий вектор,
- значение градиента в точке M.

Найдем длину вектора $\vec{c}$

$$|\overrightarrow{\mathrm{c}}| = \sqrt{x_0^2 + y_0^2 + z_0^2} = \sqrt{(-9)^2 + 8^2(-12)^2} = \sqrt{289} = 17$$

Теперь сделаем единичный вектор

$$\overrightarrow{\mathrm{c}_0} = \frac{\overrightarrow{\mathrm{c}}}{|\overrightarrow{\mathrm{c}}|} = \left(\frac{-9}{17}, \frac{8}{17}, \frac{-12}{17}\right)$$

Теперь найдем градиент.

$$U'_x = 2x$$

$$U'_y = 2y$$

$$U'_z = 2z$$

Тогда градиент нашей функции в точке $M(8, -12, 9)$ равен:

$$\mathrm{grad}U = (16, -24, 18)$$

Осталось посчитать скалярное произведение. Т.к. координаты в нашем случае известны, то достаточно перемножить координаты и сложить результат.

$$U'_{\overrightarrow{\mathrm{c}}} = \frac{-9}{17} \cdot 16 + \frac{8}{17} \cdot -24 + \frac{-12}{17} \cdot 18 = -\frac{144}{17} - \frac{192}{17} - \frac{216}{17} = \frac{552}{17} - 32\frac{8}{17}$$

4. Найти производную функции $U = e^{x^2+y^2+z^2}$ по направлению вектора $\overrightarrow{c}(4, -13, -16)$ в точке $L(-16, 4, -13)$

Для нахождения производной по направлению нам необходимо знать:

- единичный направляющий вектор,
- значение градиента в точке L.

Найдем длину вектора $\overrightarrow{c}$

$$|\overrightarrow{\mathrm{c}}| = \sqrt{x_0^2 + y_0^2 + z_0^2} = \sqrt{(4)^2 + (-13)^2 + (-16)^2} = \sqrt{441} = 21$$

Теперь сделаем единичный вектор

$$\overrightarrow{c_0} = \frac{\overrightarrow{c}}{|\overrightarrow{c}|} = \left(\frac{4}{21}, -\frac{13}{21}, -\frac{16}{21}\right)$$

Теперь найдем градиент.

$$U'_x = 2x \cdot e^{x^2+y^2+z^2}$$

$$U'_y = 2y \cdot e^{x^2+y^2+z^2}$$

$$U'_z = 2z \cdot e^{x^2+y^2+z^2}$$

Тогда градиент нашей функции в точке $L(-16, 4, -13)$ равен:

$$\text{grad}U = (-32e^{441}, 8e^{441}, -26e^{441})$$

Осталось посчитать скалярное произведение. Т.к. координаты в нашем случае известны, то достаточно перемножить координаты и сложить результат.

$$U'_{\overrightarrow{c}} = -\frac{4}{21} \cdot 32e^{441} - \frac{13}{21} \cdot 8e^{441} + \frac{16}{21} \cdot 26e^{441} = -\frac{128e^{441}}{21} - \frac{104e^{441}}{21} + \frac{416}{2}$$

5*. Найти производную функции $U = \log_{21}(x^2 + y^2 + z^2)$ в точке $F(-19, 8, -4)$ по самому быстрому направлению.

Для нахождения производной по направлению нам необходимо знать:

- единичный направляющий вектор,
- значение градиента в точке L.

В данном случае самое быстрое направление изменения функции это и будет направление градиента. Градиент и будет направляющим вектором. Найдем его в точке F(-19,8,-4).

Теперь найдем градиент.

$$U'_x = \frac{2x}{\ln 21 \cdot (x^2 + y^2 + z^2)}$$

$$U'_y = \frac{2y}{\ln 21 \cdot (x^2 + y^2 + z^2)}$$

$$U_z' = \frac{2z}{\ln 21 \cdot (x^2 + y^2 + z^2)}$$

Тогда градиент нашей функции в точке $F(-19, 8, -4)$ будет описываться вектором:

$$\mathrm{grad}U = (-\frac{38}{441 \cdot \ln 21}, \frac{16}{441 \cdot \ln 21}, -\frac{8}{441 \cdot \ln 21})$$

Найдем его длинну

$$|\mathrm{grad}U| = \sqrt{\left(-\frac{38}{441 \cdot \ln 21}\right)^2 + \left(\frac{16}{441 \cdot \ln 21}\right)^2 + \left(-\frac{8}{441 \cdot \ln 21}\right)^2}$$
$$= \sqrt{\frac{38^2}{(441 \cdot \ln 21)^2} + \frac{16^2}{(441 \cdot \ln 21)^2} + \frac{8^2}{(441 \cdot \ln 21)^2}} = \frac{\sqrt{1764}}{441 \cdot \ln 21} = \frac{42}{441 \cdot \ln 2}$$

Теперь сделаем единичный вектор

$$\mathrm{grad}U_0 = \frac{\mathrm{grad}U}{|\mathrm{grad}U|} = \left(-\frac{38}{42}, \frac{16}{42}, -\frac{8}{42}\right)$$

Осталось посчитать скалярное произведение. Т.к. координаты в нашем случае известны, то достаточно перемножить координаты и сложить результат.

$$U'_{\mathrm{grad}U} = \left(\frac{38}{42} \cdot \frac{38}{441 \cdot \ln 21} + \frac{16}{42} \cdot \frac{16}{441 \cdot \ln 21} + \frac{8}{42} \cdot \frac{8}{441 \cdot \ln 21}\right) = \frac{176}{18522 \cdot}$$
$$= \frac{2}{21 \cdot \ln 21}$$

Видим, что значение производной по самому быстрому направлению совпадает с длинной направляющего вектора. Тоже самое было и на вебинаре. Это наверное и есть тот более простой способ определения этой производной :)

6. Исследовать на экстремум функцию:

$$U = x^2 y + \frac{1}{3} y^3 + 2x^2 + 3y^2 - 1$$

Найдем частные производные, приравняем их к нулю и найдем стационарные точки:

$$\begin{cases} U'_x = 2xy + 4x = 0, \\ U'_y = x^2 + y^2 + 6y = 0 \end{cases}$$

При использовании сервиса wolframalpha удалось найти следующие корни данной системы:

$$x_1 = 0,\ y_1 = -6$$

$$x_2 = 0,\ y_2 = 0$$

$$x_3 = -2\sqrt{2},\ y_3 = -2$$

$$x_4 = 2\sqrt{2},\ y_4 = -2$$

Теперь найдем вторые производные и составим матрицу Гёссе в общем виде:

$$U''_{xx} = 2y + 4$$
$$U''_{xy} = U''_{yx} = 2x$$
$$U''_{yy} = 2y + 6$$

Видно, что все 4 значения матрицы завиясят от переменных

$$\begin{pmatrix} 2y + 4 & 2x \\ 2x & 2y + 6 \end{pmatrix}$$

теперь можно вычислить главные миноры

$$\Delta_1 = U''_{xx} = 2y + 4$$

$$\Delta_2 = \begin{vmatrix} U''_{xx} & U''_{xy} \\ U''_{yx} & U''_{yy} \end{vmatrix} = U''_{xx} \cdot U''_{yy} - (U''_{xy})^2 = 4y^2 + 20y + 24 - 4x^2$$

Осталось проверить знаки миноров для каждой точки.

In [6]:

```
roots = pd.DataFrame([
    ["Точка 1", 0, -6],
    ["Точка 2", 0, 0],
    ["Точка 3", -2*sqrt(2), -2],
    ["Точка 4", 2*sqrt(2), -2]
])
for r in range(len(roots)):
    minor_1 = 2*roots.loc[r][2]+4
    minor_2 = 4*(roots.loc[r][2]**2)+20*roots.loc[r][2]+24-4*(roots.loc[r][1]**2)
    print("minor_1 =",minor_1, "," "minor_2 =",minor_2)
    if minor_1 > 0 and minor_2 > 0:
        print(f"Стационарная точка '{roots.loc[r][0]}' с координатами {roots.loc[r][1], roo
    elif minor_1 < 0 and minor_2 > 0:
        print(f"Стационарная точка '{roots.loc[r][0]}' с координатами {roots.loc[r][1], roo
    elif minor_1 == 0 or minor_2 == 0:
        print(f"Экстремум для точки '{roots.loc[r][0]}' с координатами {roots.loc[r][1], ro
    else:
        print(f"Стационарная точка '{roots.loc[r][0]}' с координатами {roots.loc[r][1], roo
    print()
```

```
minor_1 = -8 ,minor_2 = 48
Стационарная точка 'Точка 1' с координатами (0, -6) является максимумом

minor_1 = 4 ,minor_2 = 24
Стационарная точка 'Точка 2' с координатами (0, 0) является минимумом

minor_1 = 0 ,minor_2 = -32
Экстремум для точки 'Точка 3' с координатами (-2*sqrt(2), -2) с помощью теори
и матрицы Гессе найти не возможно

minor_1 = 0 ,minor_2 = -32
Экстремум для точки 'Точка 4' с координатами (2*sqrt(2), -2) с помощью теории
матрицы Гессе найти не возможно
```

Построим график

In [7]:

```
import pylab
from mpl_toolkits.mplot3d import Axes3D
import numpy

def makeData ():
    x = numpy.arange (-10, 10, 0.01)
    y = numpy.arange (-10, 10, 0.01)
    xgrid, ygrid = numpy.meshgrid(x, y)

    Ugrid = xgrid**2*y+1/3*ygrid**3+2*xgrid**2+3*ygrid**2-1
    return xgrid, ygrid, Ugrid

x, y, U = makeData()

fig = pylab.figure(figsize=(10, 10))
axes = Axes3D(fig)
# axes.view_init(10, 0)

axes.plot_surface(x, y, U)

pylab.show()
```

```
<Figure size 1000x1000 with 1 Axes>
```

7*. Исследовать на экстремум функцию:

$$U = e^{-\frac{x}{2}} \cdot (x^2 + y^2)$$

Найдем частные производные, приравняем их к нулю и найдем стационарные точки:

$$U' = (e^{-\frac{x}{2}})' \cdot (x^2 + y^2) + e^{-\frac{x}{2}} \cdot (x^2 + y^2)'$$

$$\begin{cases} U_x' = 2xe^{-\frac{x}{2}} - \frac{e^{-\frac{x}{2}} \cdot (x^2+y^2)}{2} \\ U_y' = 2ye^{-\frac{x}{2}} = 0 \end{cases}$$

При использовании сервиса wolframalpha удалось найти следующие корни данной системы:

$$x_1 = 0,\ y_1 = 0$$

$$x_2 = 4,\ y_2 = 0$$

Теперь найдем вторые производные и составим матрицу Гёссе в общем виде:

$$U''_{xx} = 2e^{-\frac{x}{2}} - 2xe^{-\frac{x}{2}} - \frac{e^{-\frac{x}{2}} \cdot (-\frac{x^2}{2} - \frac{y^2}{2})}{2}$$

$$U''_{xy} = U''_{yx} = -ye^{-\frac{x}{2}}$$
$$U''_{yy} = 2e^{-\frac{x}{2}}$$

Видно, что все 4 значения матрицы завиясят от переменных

$$\begin{pmatrix} 2e^{-\frac{x}{2}} - 2xe^{-\frac{x}{2}} - \frac{e^{-\frac{x}{2}} \cdot (-\frac{x^2}{2} - \frac{y^2}{2})}{2} & -ye^{-\frac{x}{2}} \\ -ye^{-\frac{x}{2}} & 2e^{-\frac{x}{2}} \end{pmatrix}$$

теперь можно вычислить главные миноры

$$\Delta_1 = U''_{xx} = 2e^{-\frac{x}{2}} - 2xe^{-\frac{x}{2}} - \frac{e^{-\frac{x}{2}} \cdot (-\frac{x^2}{2} - \frac{y^2}{2})}{2}$$

$$\Delta_2 = \begin{vmatrix} U''_{xx} & U''_{xy} \\ U''_{yx} & U''_{yy} \end{vmatrix} = U''_{xx} \cdot U''_{yy} - (U''_{xy})^2 = \Big((2e^{-\frac{x}{2}} - 2xe^{-\frac{x}{2}} - \frac{e^{-\frac{x}{2}} \cdot (-\frac{x^2}{2} - \frac{y^2}{2}}{2}$$

Осталось проверить знаки миноров для каждой точки.

In [8]:

```
roots = pd.DataFrame([
    ["Точка 1", 0, 0],
    ["Точка 2", 4, 0]
])

for r in range(len(roots)):
    x = roots.loc[r][1]
    y = roots.loc[r][2]
    minor_1 = 2*exp(-x/2)-2*x*exp(-x/2)-(exp(-x/2)*(-x**2/2-y**2/2))/2
    minor_2 = minor_1*2*exp(-x/2)-(y*exp(-x/2))**2
    print("minor_1 =",minor_1, ","  "minor_2 =",minor_2)
    if minor_1 > 0 and minor_2 > 0:
        print(f"Стационарная точка '{roots.loc[r][0]}' с координатами {roots.loc[r][1], roo
    elif minor_1 < 0 and minor_2 > 0:
        print(f"Стационарная точка '{roots.loc[r][0]}' с координатами {roots.loc[r][1], roo
    elif minor_1 == 0 or minor_2 == 0:
        print(f"Экстремум для точки '{roots.loc[r][0]}' с координатами {roots.loc[r][1], ro
    else:
        print(f"Стационарная точка '{roots.loc[r][0]}' с координатами {roots.loc[r][1], roo
    print()
```

```
minor_1 = 2.00000000000000 ,minor_2 = 4.00000000000000
Стационарная точка 'Точка 1' с координатами (0, 0) является минимумом

minor_1 = -0.270670566473225 ,minor_2 = -0.0732625555549367
Стационарная точка 'Точка 2' с координатами (4, 0) является седловой
```

8.** С помощью метода наименьших квадратов (МНК) подобрать значения параметров a и b для сигмоидальной функции

$$y = \frac{1}{1 + e^{-(ax+b)}}$$

по заданному распределению. Построить график.

Представим уравнение в неявном виде

$$y - \frac{1}{1 + e^{-(ax+b)}} = 0$$

Найдем частные производные

In [9]:

```
xi, yi, a, b =symbols('xi yi a b')
f=(yi-1/(1+math.e**(-a*xi-b)))**2
f
```

Out[9]:

$$\left(yi - \frac{1}{2.71828182845905^{-a\xi-b} + 1}\right)^2$$

In [10]:

```
print(f"Производная первого порядка по x: ", diff(f,xi))
```

```
Производная первого порядка по x:  -2.0*2.71828182845905**(-a*xi - b)*a*(yi -
1/(2.71828182845905**(-a*xi - b) + 1))/(2.71828182845905**(-a*xi - b) + 1)**2
```

In [11]:

```
print(f"Производная первого порядка по y: ", diff(f,yi))
```

```
Производная первого порядка по y:  2*yi - 2/(2.71828182845905**(-a*xi - b) +
1)
```

In [12]:

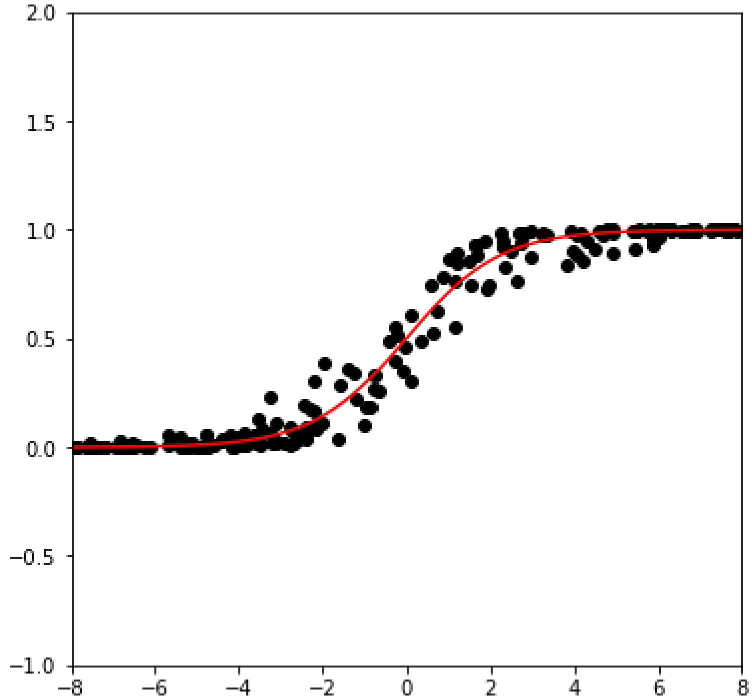
```
from matplotlib import pylab as plt
import numpy as np
%matplotlib inline
from scipy.optimize import fsolve
import math


xi = np.random.uniform(-8, 8, 200)
yi = 1/(1+np.exp(-np.random.uniform(0.5, 1.5, 200)*xi-np.random.uniform(-1, 1, 200)))
```

Найдем коэффициенты и построим гарфики

In [21]:

```
plt.figure(figsize=(6,6))
plt.axis([-8, 8, -1, 2])

def equations(p):
    a, b = p
    # Запись систем уравнения
    return ((-2*a*(yi - 1/(math.e**(-a*xi - b) + 1))*math.e**(-a*xi - b)/(math.e**(-a*xi -

# Численное решение системы уравнений
a, b =  fsolve(equations, (1, 1))
print (a, b)

x = np.linspace(-8, 8, 200)
y=1/(1+2.71828182845905**(-a*x-b))

plt.scatter(xi,yi, c='black')
plt.plot(x,y, c='r')
plt.show()
```

```
0.8983039614455821 0.007196287949449307
```



In [22]:

```
y_=y.sum()/200
r=np.sqrt((((yi-y_)**2).sum()-((yi-1/(1+2.71828182845905**(-a*xi-b)))**2).sum())/((yi-y_)**

print (r)
```

```
0.9895722674929515
```

In [23]:

```
from sklearn.metrics import mean_squared_error
from math import sqrt

rms = sqrt(mean_squared_error(yi, y))
rms
```

Out[23]:

0.5604560448019348

In [24]:

```
from sklearn.metrics import mean_absolute_error
mean_absolute_error(yi, y)
```

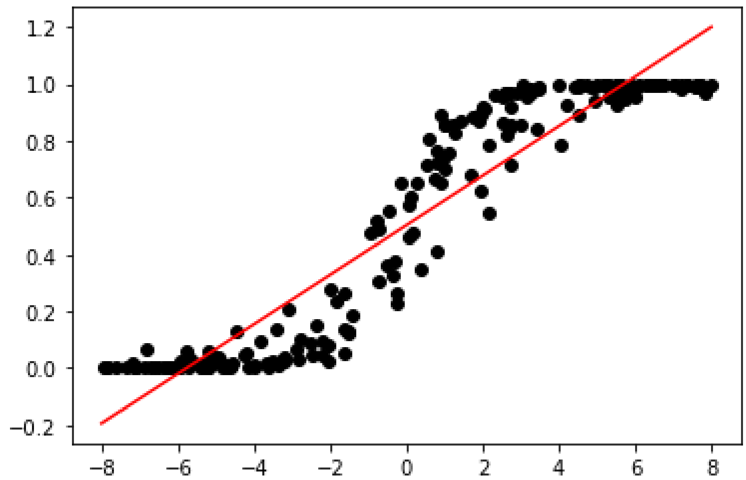
Out[24]:

0.41071092583453006

In [25]:

```
xi = np.random.uniform(-8, 8, 200)
yi = 1/(1+np.exp(-np.random.uniform(0.5, 1.5, 200)*xi-np.random.uniform(-1, 1, 200)))

def equations(p):
    k, b = p
    # Запись системы уравнения вида x+y=2, x-y=0
    return ((-2*xi*(yi-k*xi-b)).sum(), -2*(yi-k*xi-b).sum())

# Численное решение системы уравнений
k, b =  fsolve(equations, (0, 0))
print (k, b)

x = np.linspace(-8, 8, 200)
y=k*x+b

plt.scatter(xi,yi, c='black')
plt.plot(x,y, c='r')
plt.show()
```

```
0.08718844955716l4 0.5030273666927758
```



In [26]:

```
y_=y.sum()/200
r=np.sqrt((((yi-y_)**2).sum()-((yi-1/(1+2.71828182845905**(-a*xi-b)))**2).sum())/((yi-y_)**

print (r)
```

0.9789992213860433

In [27]:

```
from sklearn.metrics import mean_squared_error
from math import sqrt

rms = sqrt(mean_squared_error(yi, y))
rms
```

Out[27]:

0.5933475468435067

In [28]:

```
from sklearn.metrics import mean_absolute_error
mean_absolute_error(yi, y)
```

Out[28]:

0.4886085829396531

Видим, что корреляция, среднеквадратичная ошика и средняя абсолютная ашибка не сильно подходит для оценки качества аппроксимации в данном случае