**Московский государственный технический университет им. Н.Э. Баумана.**

Факультет «Информатика и системы управления»

Кафедра «Системы обработки информации и управления»

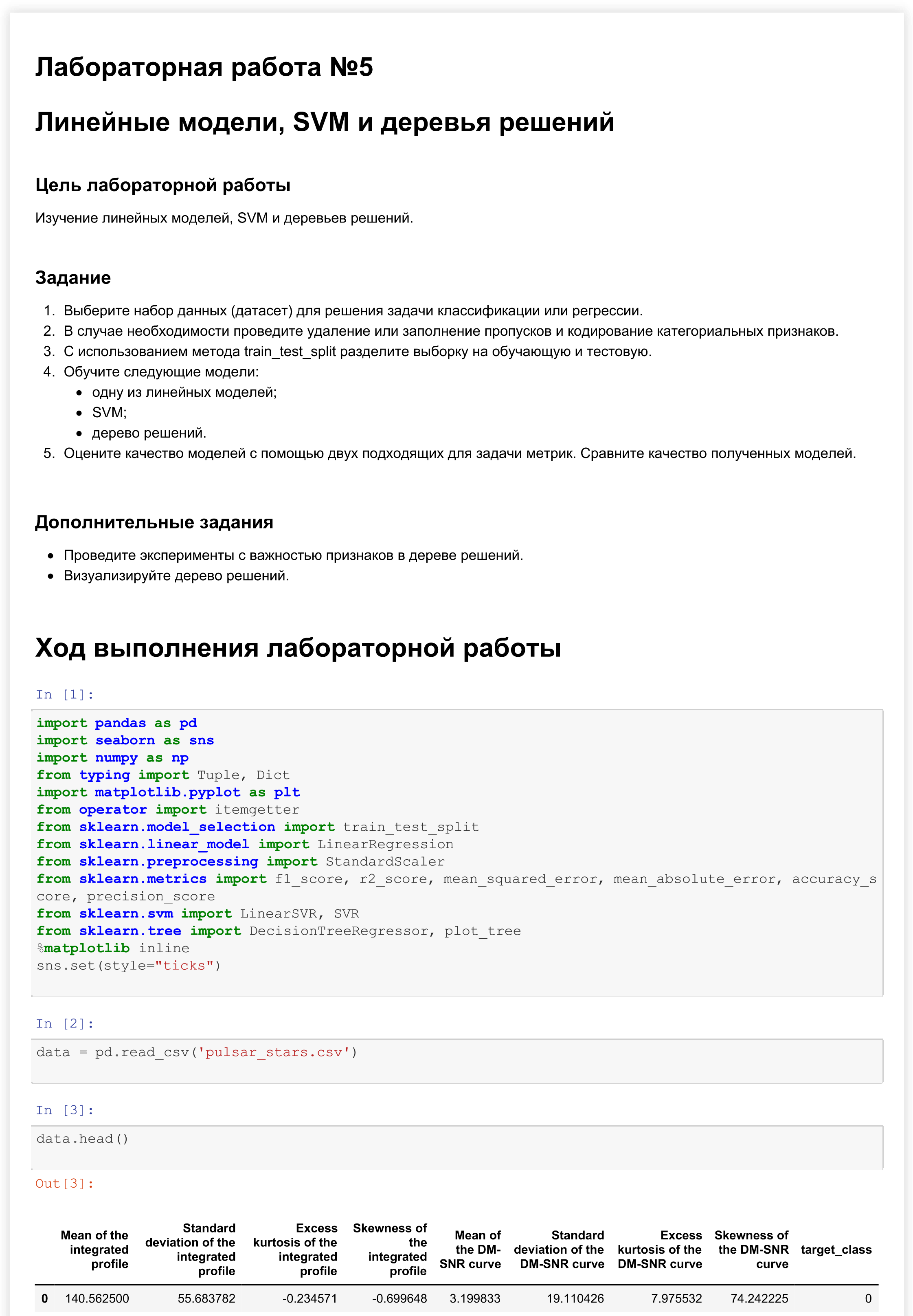
# Курс «Технологии машинного обучения»

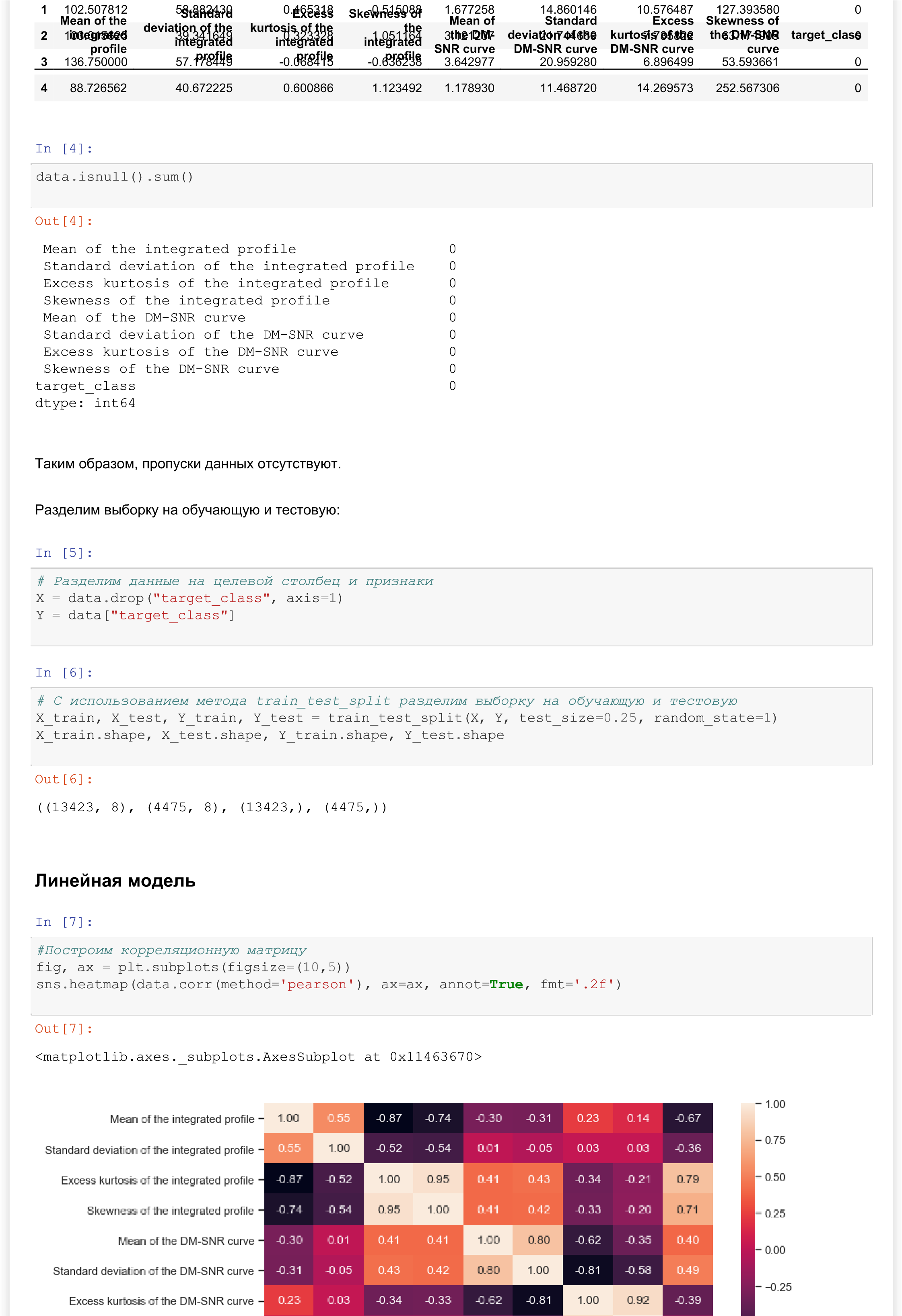
Отчет по лабораторной работе №5

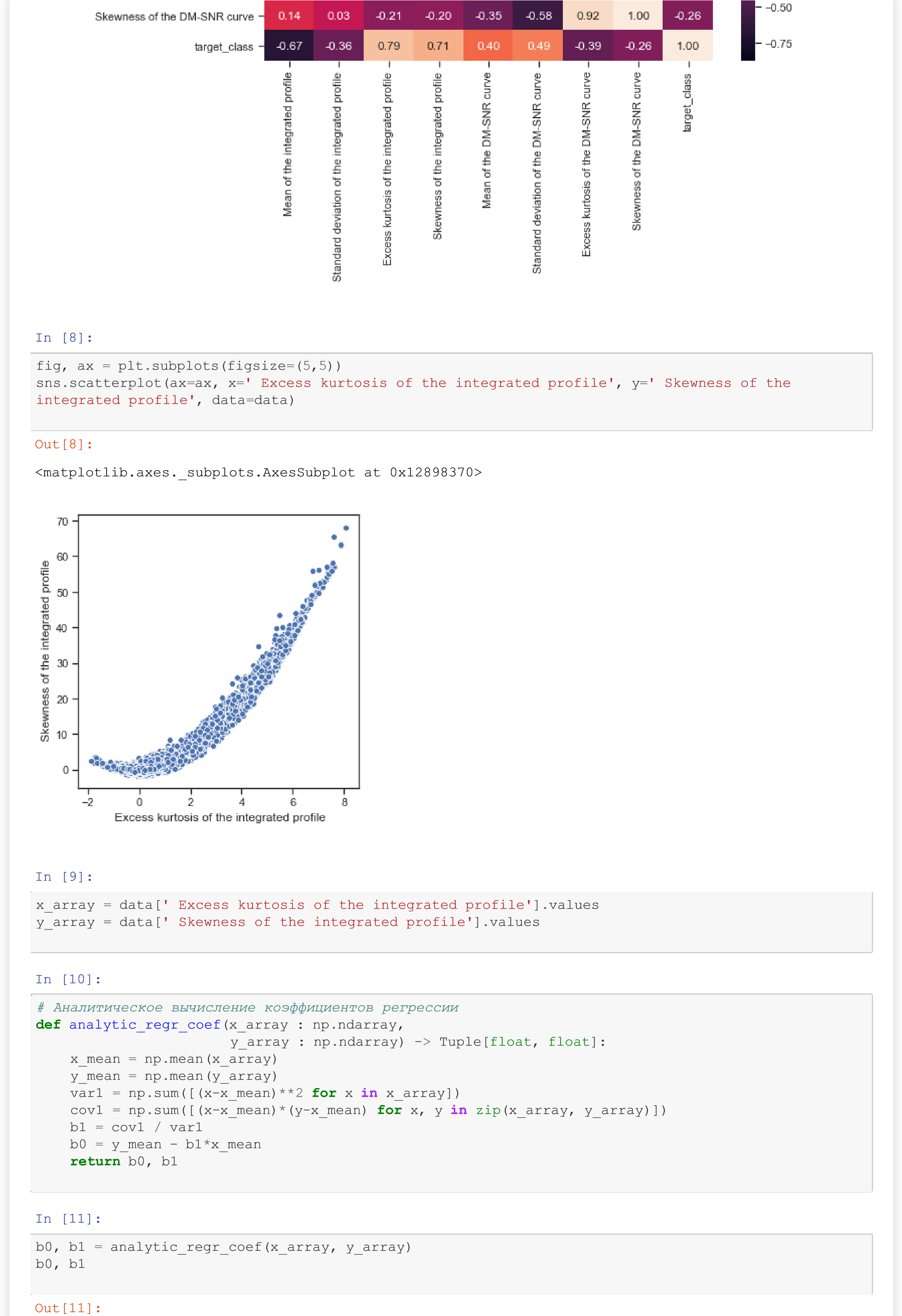
Выполнил: Проверил:

студент группы ИУ5-62Б преподаватель каф. ИУ5   
Грачев Я.А. Гапанюк Ю.Е. Подпись и дата: Подпись и дата:

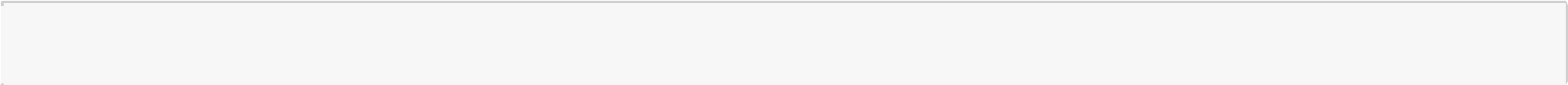
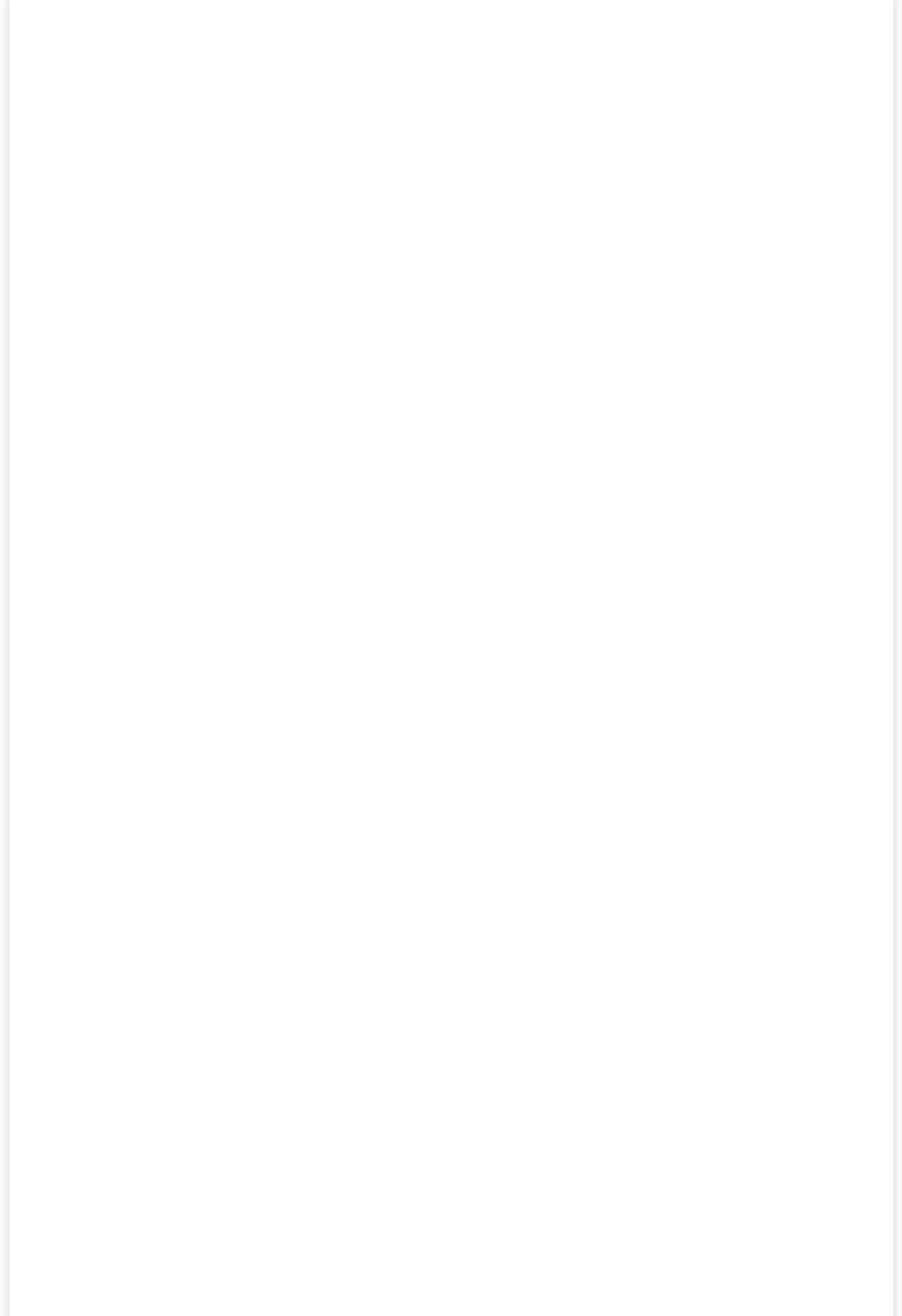
Москва, 2020 г.











In [17]:

show\_gradient\_descent

(

1000

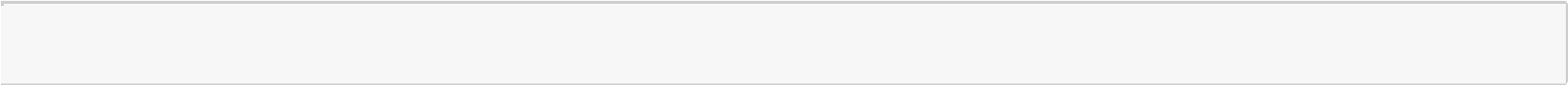
,

0

,

0

)



In [18]:

show\_gradient\_descent

(

10000

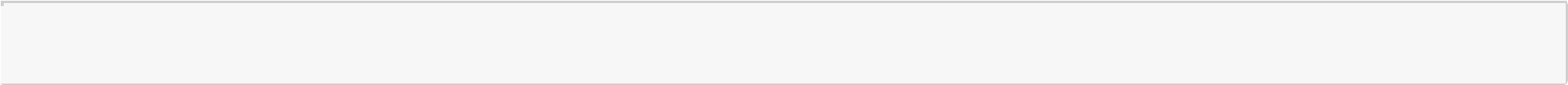
,

0

,

0

)



In [19]:

show\_gradient\_descent

(

10

,

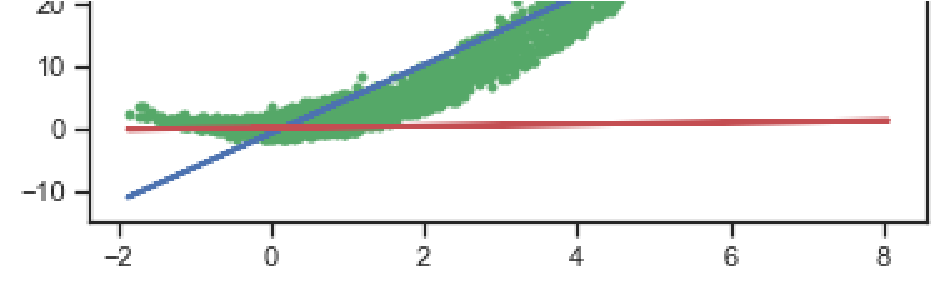
-

2

,

5

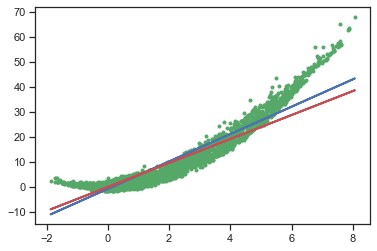
)



b0 = -0.8493834635342365 - (теоретический), -0.1046956764421704 - (градиентный спуск)

b1 = 5.482102483932346 - (теоретический), 4.8016950489981 - (градиентный спуск)

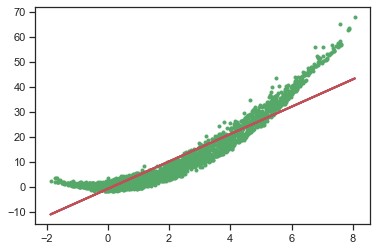
MSE = 0.700140861729278



b0 = -0.8493834635342365 - (теоретический), -0.8493787037369952 - (градиентный спуск)

b1 = 5.482102483932346 - (теоретический), 5.482099192076996 - (градиентный спуск)

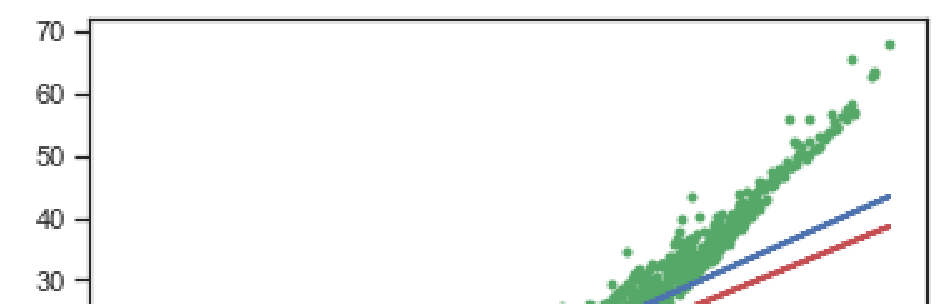
MSE = 2.2423416471452288e-11

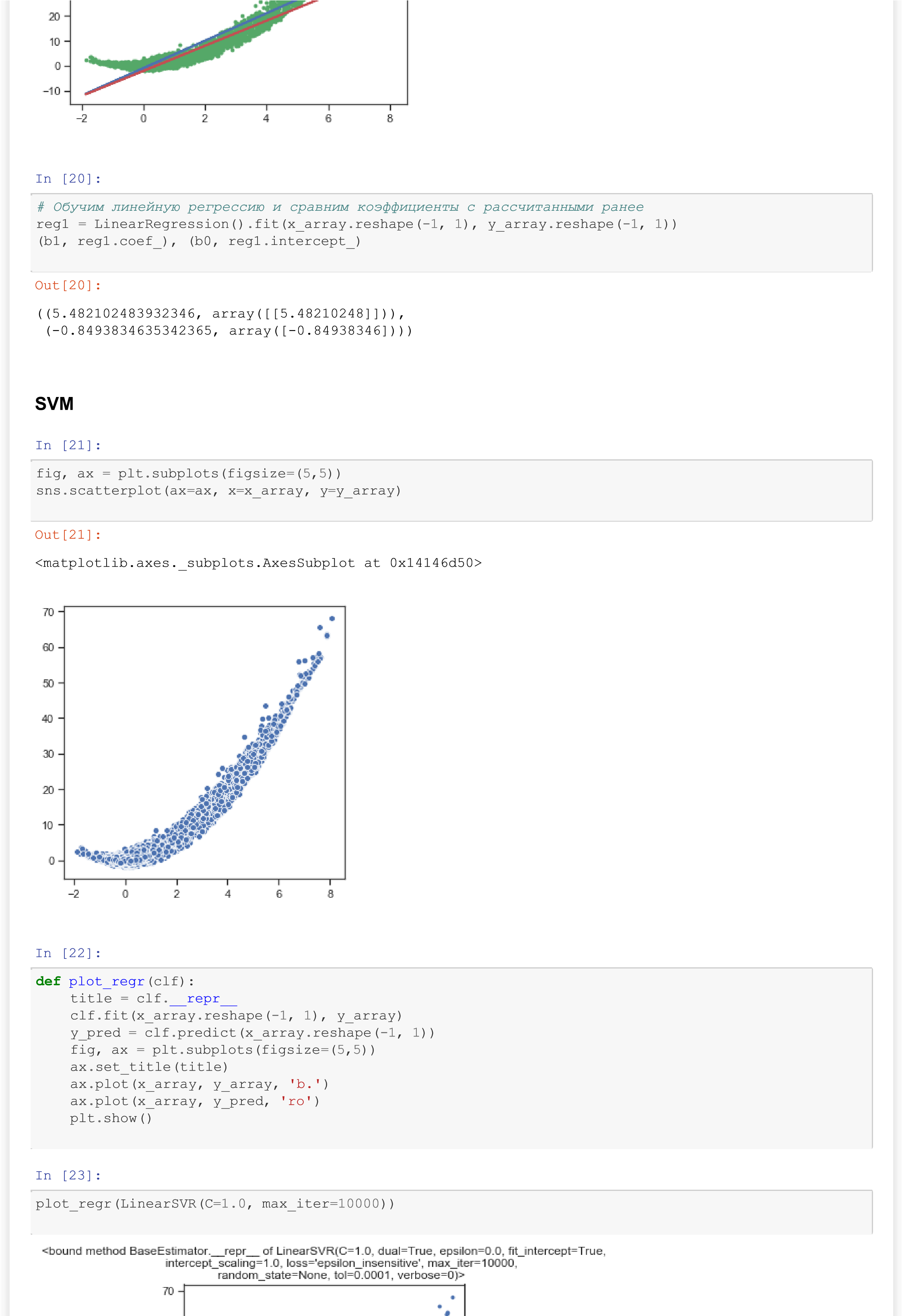


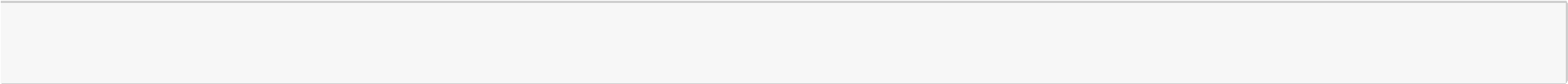
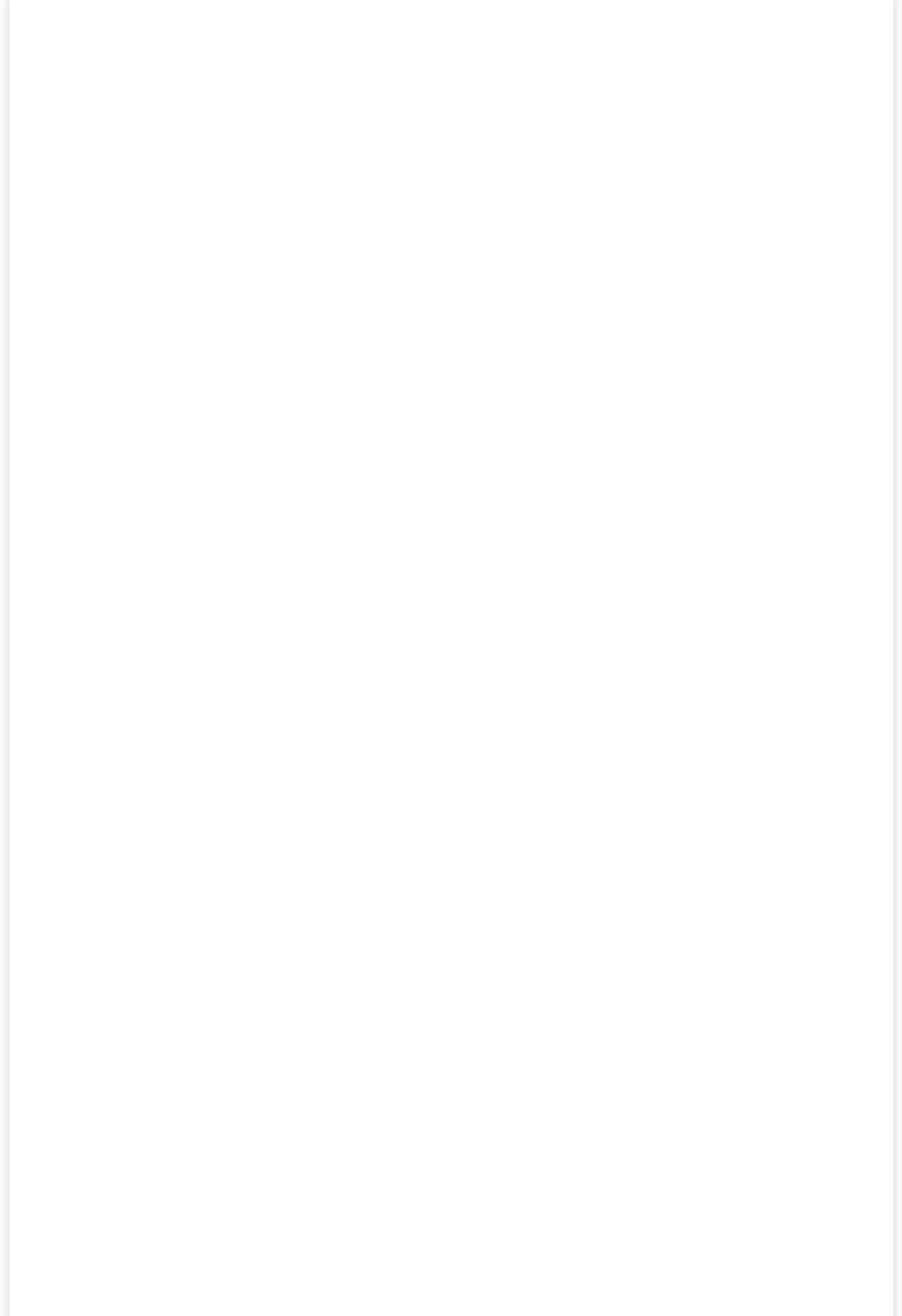
b0 = -0.8493834635342365 - (теоретический), -1.972729515823501 - (градиентный спуск)

b1 = 5.482102483932346 - (теоретический), 5.023704112532512 - (градиентный спуск)

MSE = 2.039915190146421







In [24]:

plot\_regr

(

SVR

(

kernel

=

'rbf'

,

gamma

=

0.2

,

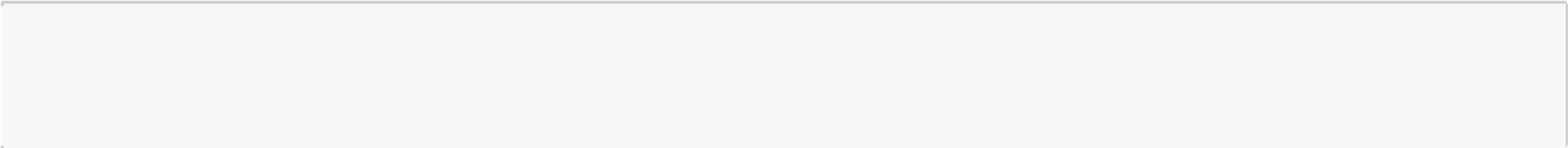
C

=

1.0

))

**Дерево решений**



In [25]:

*# Обучим дерево на всех признаках*

tree

=

DecisionTreeRegressor

(

random\_state

=

1

)

tree

.

fit

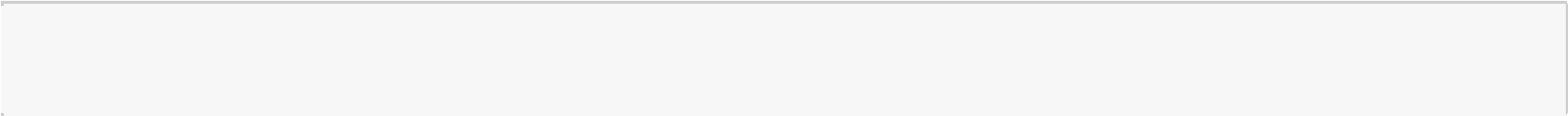
(

X\_train

,

Y\_train

)



In [26]:

*# Важность признаков*

list

(

zip

(

X\_train

.

columns

.

values

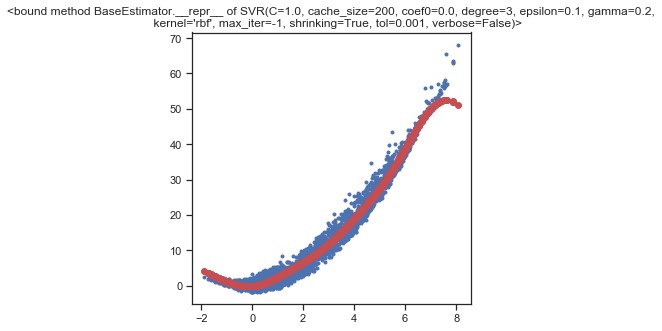
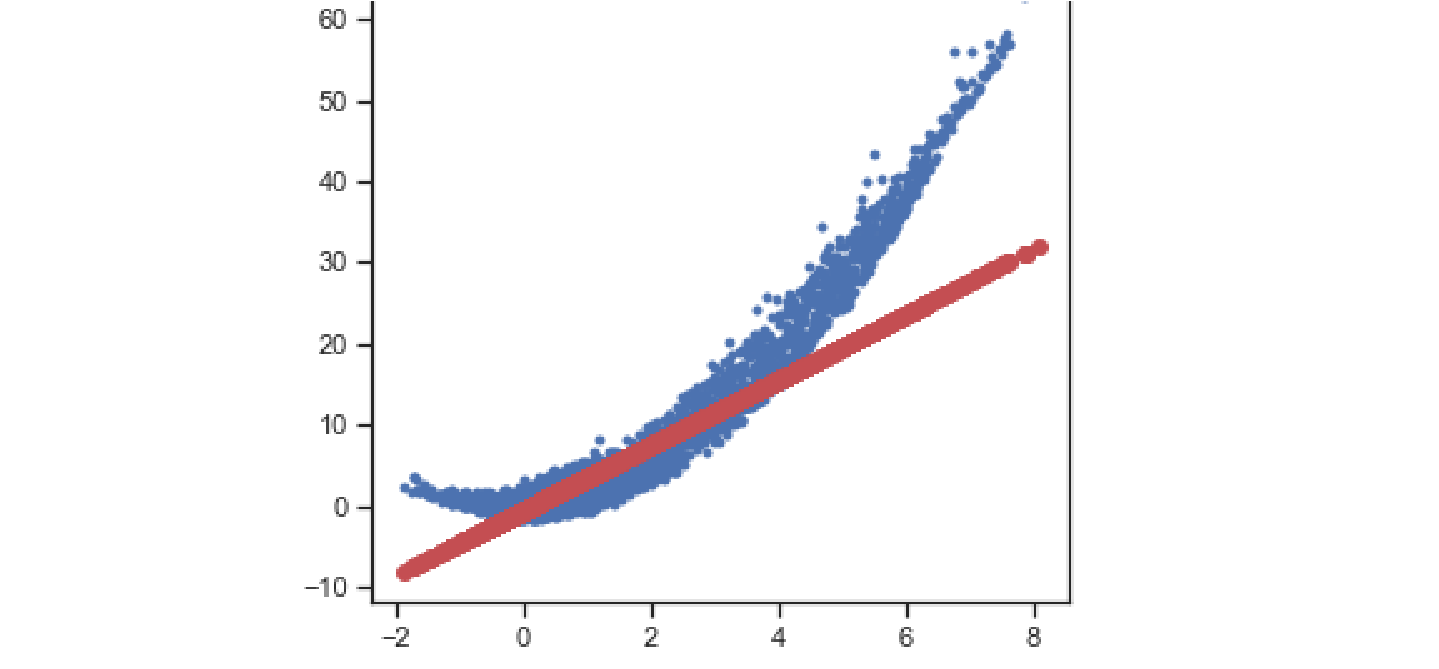
,

tree

.

feature\_importances\_

))



Out[25]:

DecisionTreeRegressor(ccp\_alpha=0.0, criterion='mse', max\_depth=None,

max\_features=None, max\_leaf\_nodes=None,

min\_impurity\_decrease=0.0, min\_impurity\_split=None,

min\_samples\_leaf=1, min\_samples\_split=2,

min\_weight\_fraction\_leaf=0.0, presort='deprecated',

random\_state=1, splitter='best')

Out[26]:

[(

' Mean of the integrated profile', 0.02764437664705047),

(' Standard deviation of the integrated profile', 0.029344493324682287),

(' Excess kurtosis of the integrated profile', 0.7986346670595349),

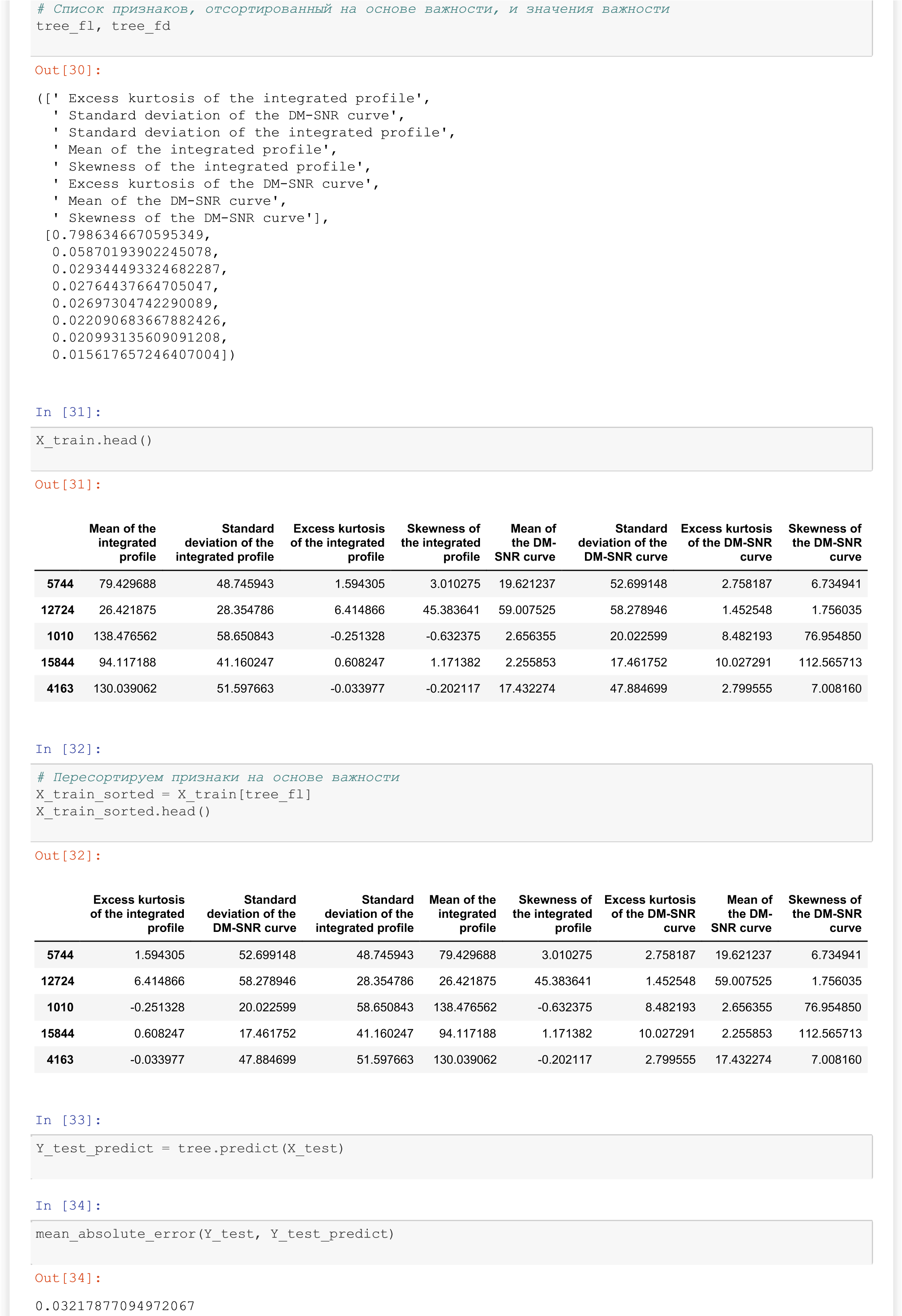
(' Skewness of the integrated profile', 0.02697304742290089),

(' Mean of the DM-SNR curve', 0.020993135609091208),

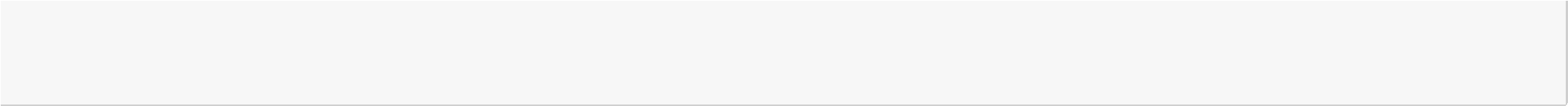
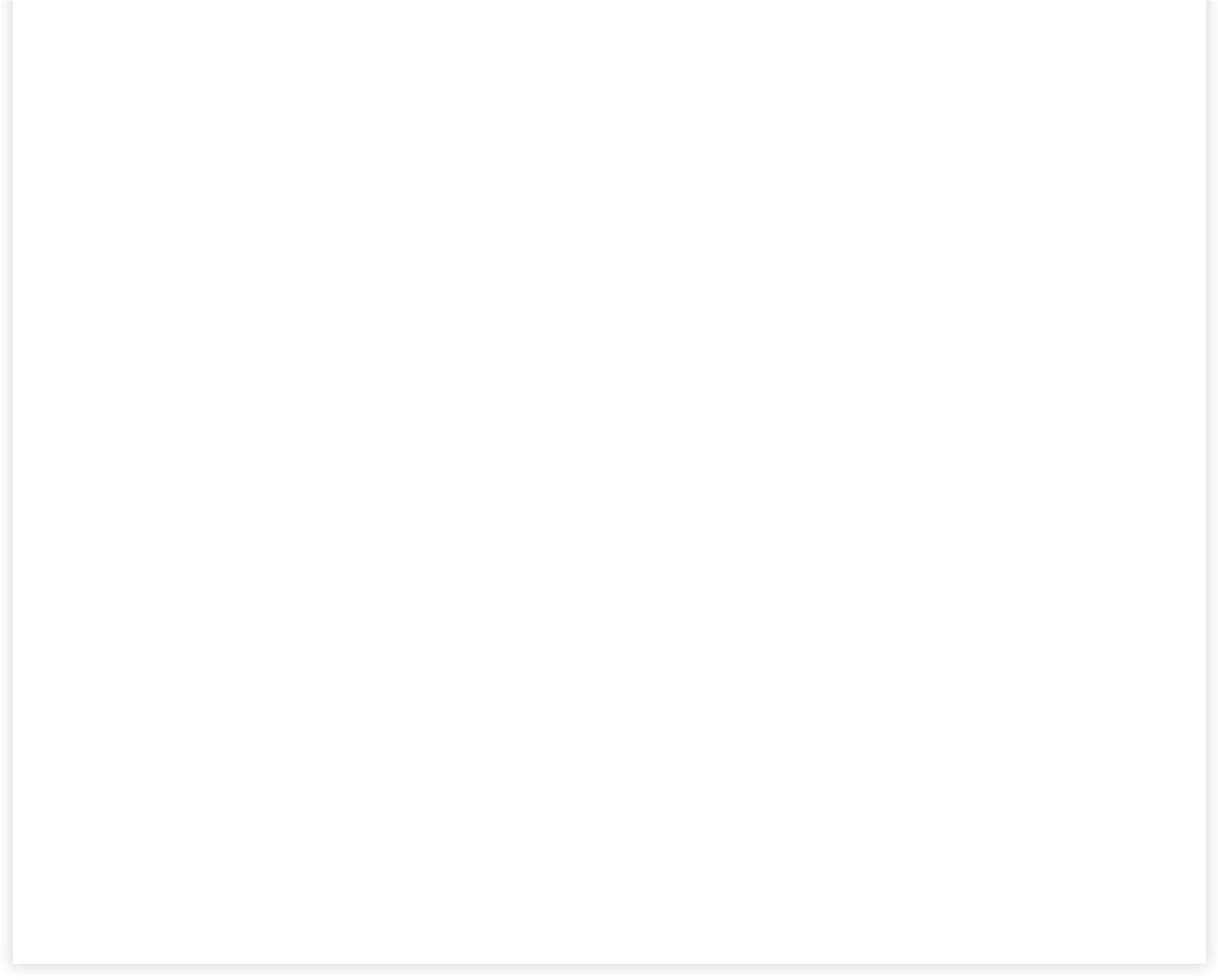
(' Standard deviation of the DM-SNR curve', 0.05870193902245078),

(' Excess kurtosis of the DM-SNR curve', 0.022090683667882426),









print

(

"r2\_score:"

,

r2\_score

(

Y\_test

,

tree

.

predict

(

X\_test

)))

print

(

"mean\_squared\_error:"

,

mean\_squared\_error

(

Y\_test

,

tree

.

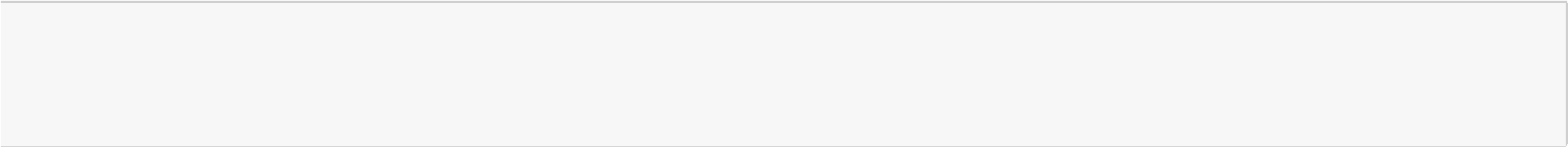
predict

(

X\_test

)))

Линейная регрессия



In [41]:

pred

=

reg1

.

predict

(

x\_array

.

reshape

(

-

1

,

1

))

print

(

"r2\_score:"

,

r2\_score

(

y\_array

,

pred

))

print

(

"mean\_squared\_error"

,

mean\_squared\_error

(

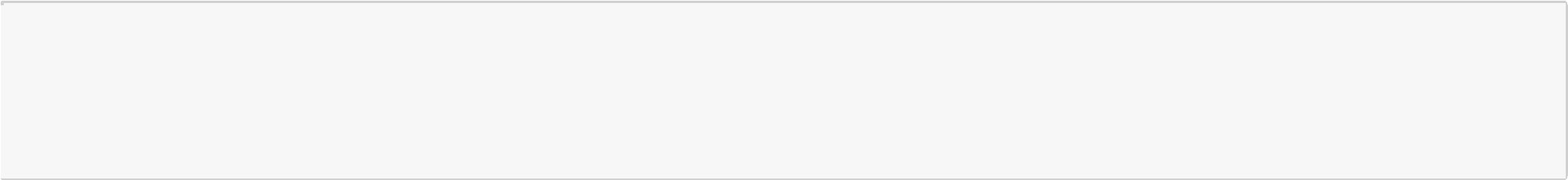
y\_array

,

pred

))

Метод опорных векторов



In [42]:

svr

=

SVR

(

kernel

=

'rbf'

)

svr

.

fit

(

X\_train

,

Y\_train

)

print

(

"r2\_score:"

,

r2\_score

(

Y\_test

,

svr

.

predict

(

X\_test

)))

print

(

"mean\_squared\_error"

,

mean\_squared\_error

(

Y\_test

,

svr

.

predict

(

X\_test

)))

Можно сказать, что все три модели являются приемлемыми, т.к. коэффициент детерминации для всех трех моделей больше

50

%.

Если учитывать показатели обеих метрик, наилучший результат показал метод опорных векторов.

r2\_score: 0.608192211999387

mean\_squared\_error: 0.03217877094972067

r2\_score: 0.8944034470145041

mean\_squared\_error 4.017001458201285

r2\_score: 0.7018891842122201

mean\_squared\_error 0.02448353491854106