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

Факультет «Информатика и системы управления»

Кафедра «Системы обработки информации и управления»

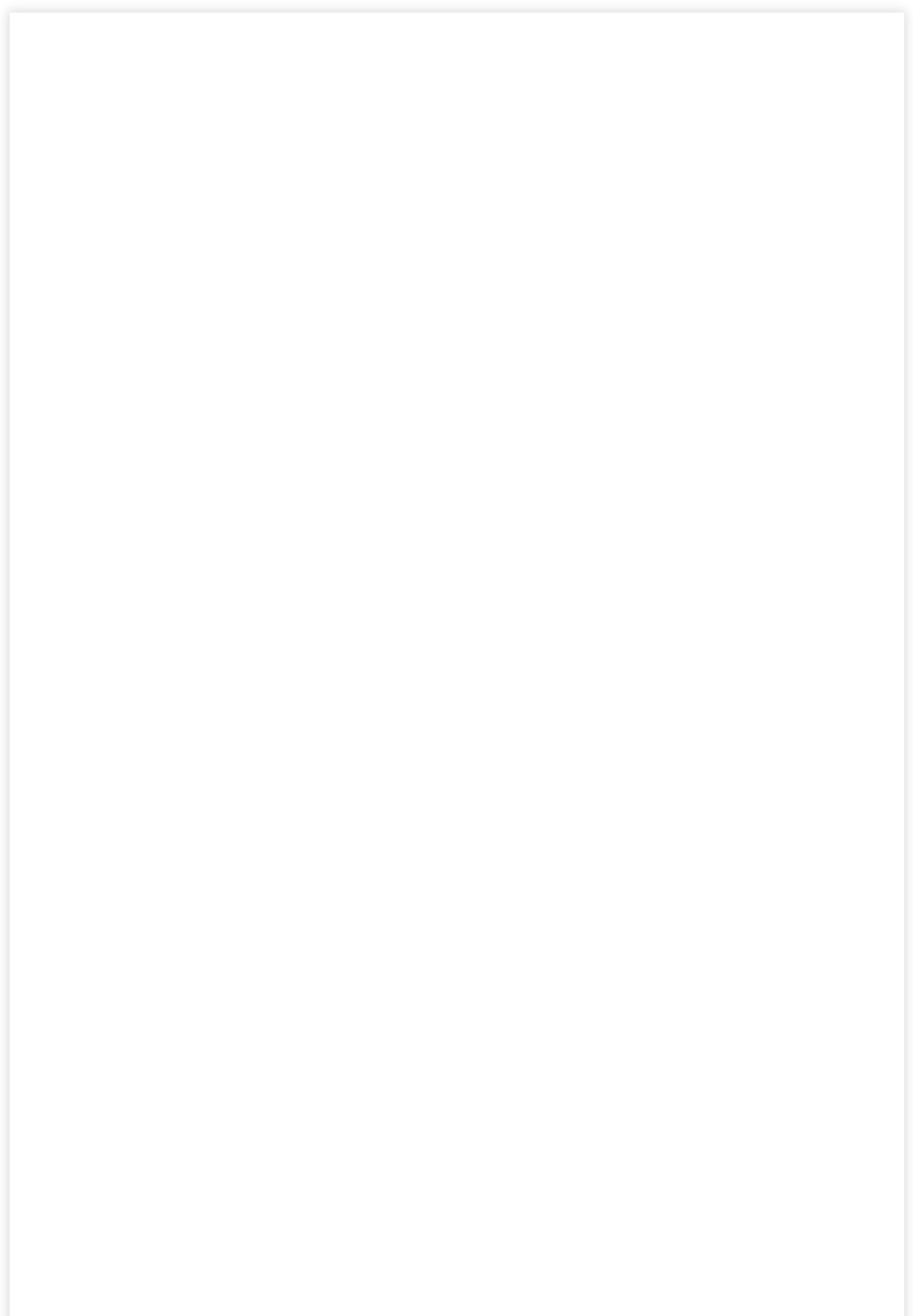
Курс «Технологии машинного обучения»

Рубежный контроль №2

Выполнил: Проверил:

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

Москва, 2020 г.



**Рубежный контроль №2**

Работа Брусникиной М.И., группа ИУ5-62Б, вариант 3

**Задача (Кластеризация данных)**

Кластеризуйте данные с помощью двух алгоритмов кластеризации. Сравните качество кластеризации с помощью

следующих метрик качества кластеризации (если это возможно для Вашего набора данных):

Adjusted Rand index

Adjusted Mutual Information

Homogeneity, completeness, V-measure

Коэффициент силуэта

Сделате выводы о том, какой алгоритм осуществляет более качественную кластеризацию на Вашем наборе данных.

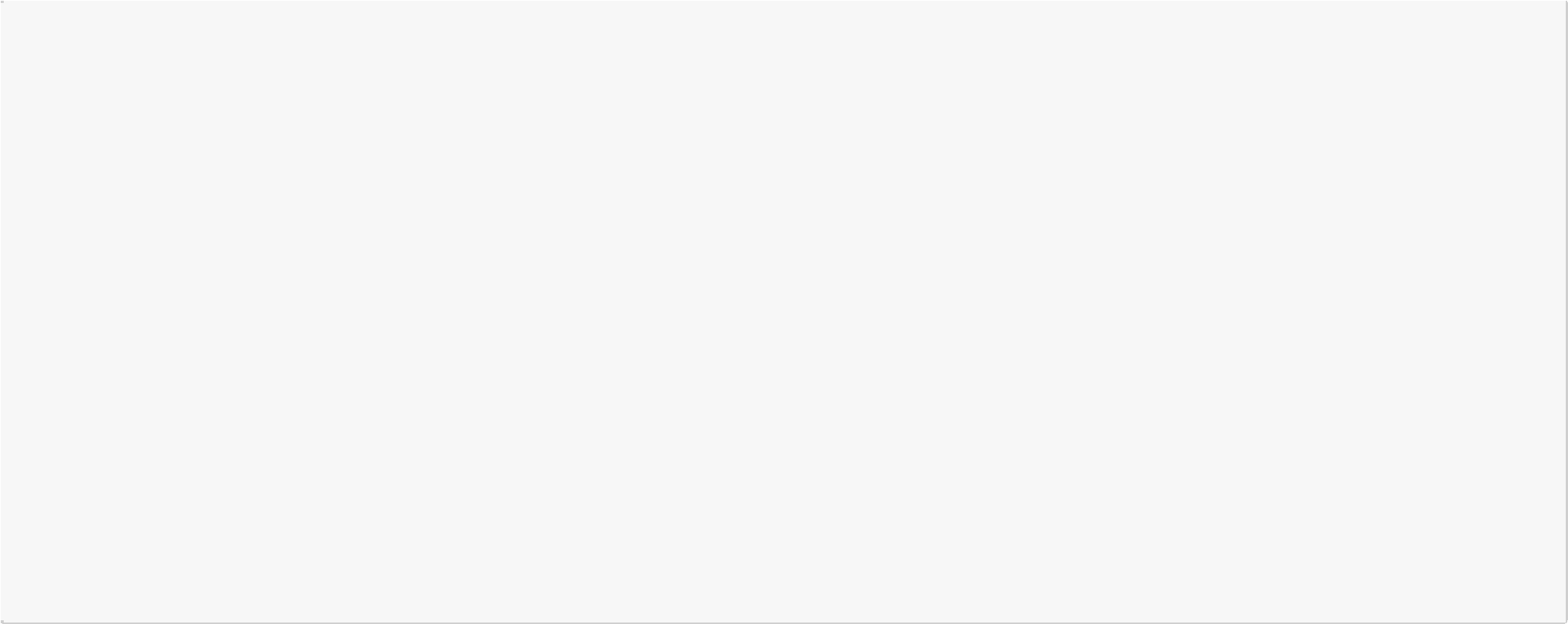
Требование для студентов группы ИУ5-62Б - использовать MeanShift и иерархическую кластеризацию.

Датасет №3 -

[e](https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_wine.html#sklearn.datasets.load_wine)

[https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load\_wine.html#sklearn.datasets.load\_win](https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_wine.html#sklearn.datasets.load_wine)

**Загрузка данных**



In [1]:

**import**

**numpy**

**as**

**np**

**import**

**pandas**

**as**

**pd**

**from**

**typing**

**import**

Dict

,

Tuple

**from**

**scipy**

**import**

stats

**from**

**sklearn**

**import**

cluster

,

datasets

,

mixture

**from**

**sklearn.neighbors**

**import**

kneighbors\_graph

**from**

**sklearn.preprocessing**

**import**

StandardScaler

**from**

**sklearn.cluster**

**import**

MeanShift

,

AgglomerativeClustering

**from**

**sklearn.metrics**

**import**

adjusted\_rand\_score

**from**

**sklearn.metrics**

**import**

adjusted\_mutual\_info\_score

**from**

**sklearn.metrics**

**import**

homogeneity\_completeness\_v\_measure

**from**

**sklearn.metrics**

**import**

silhouette\_score

**from**

**sklearn.datasets**

**import**

load\_wine

**from**

**itertools**

**import**

cycle

,

islice

**import**

**seaborn**

**as**

**sns**

**import**

**matplotlib.pyplot**

**as**

**plt**

%

**matplotlib**

inline

sns

.

set

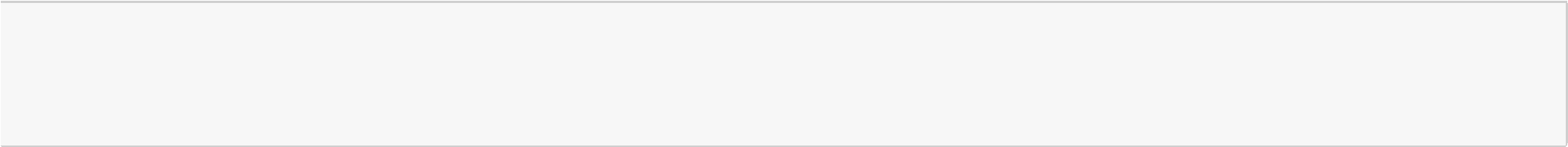
(

style

=

"ticks"

)



In [2]:

wine

=

load\_wine

()

**for**

x

**in**

wine

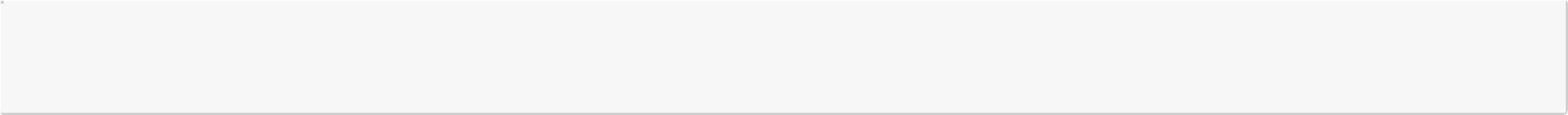
:

print

(

x

)



In [3]:

*# Признаки*

print

(

wine

.

feature\_names

)

data

target

target\_names

DESCR

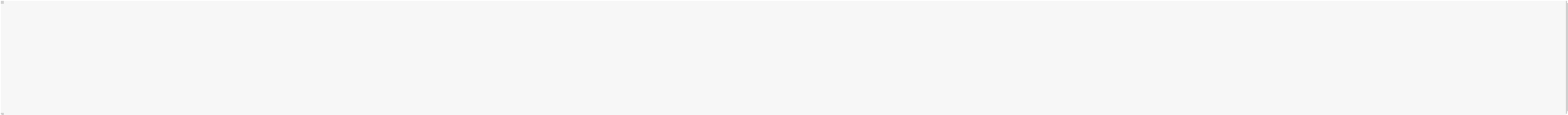
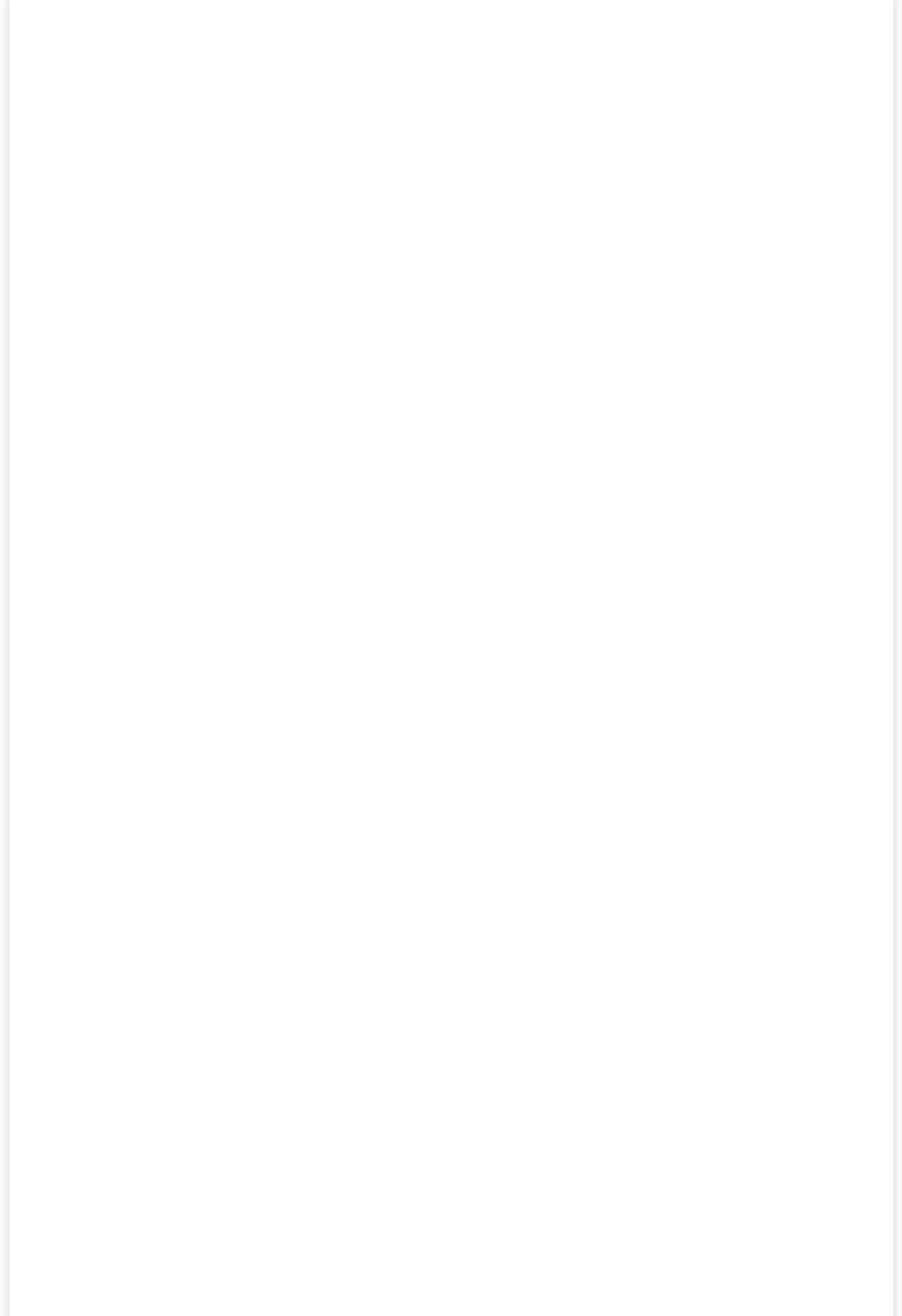
feature\_names

[

'alcohol', 'malic\_acid', 'ash', 'alcalinity\_of\_ash', 'magnesium', 'total\_phenols', 'flavanoids',

'nonflavanoid\_phenols', 'proanthocyanins', 'color\_intensity', 'hue',

'od280/od315\_of\_diluted\_wines', 'proline']



In [4]:

*# Метки*

print

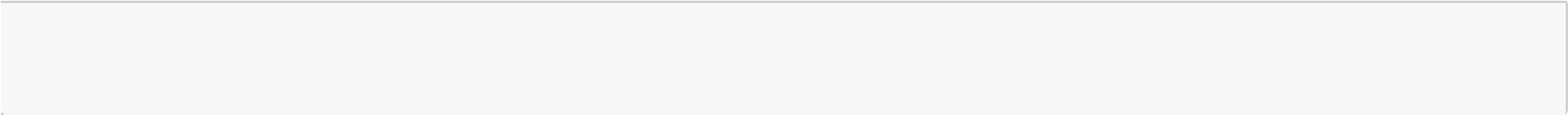
(

wine

.

target

)



In [5]:

*# Имена меток*

print

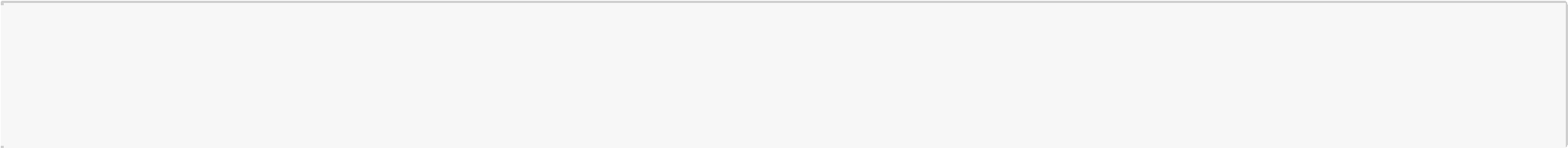
(

wine

.

target\_names

)



In [6]:

*# Разделение набора данных*

x\_axis

=

wine

.

data

,

[:

5

]

y\_axis

=

wine

.

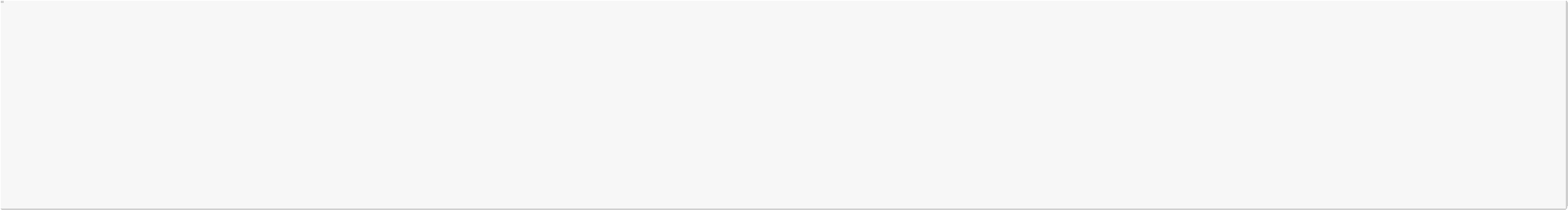
data

,

[:

9

]



In [7]:

*# Построение*

plt

.

xlabel

(

wine

.

feature\_names

[

5

])

plt

.

ylabel

(

wine

.

feature\_names

[

9

])

plt

.

scatter

(

x\_axis

,

y\_axis

,

c

=

wine

.

target

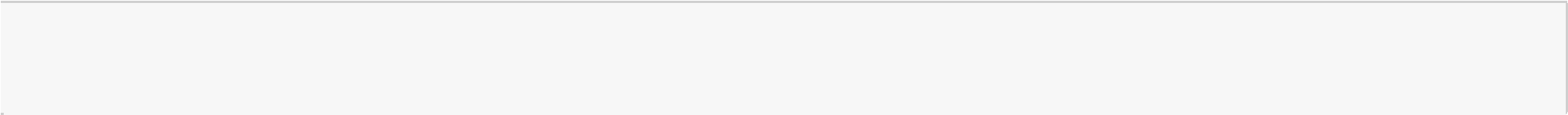
)

plt

.

show

()



In [8]:

data

=

pd

.

DataFrame

(

data

=

np

.

c\_

[

wine

.

data

[:

,

5

]

,

wine

.

data

[:

,

9

]]

,

columns

=

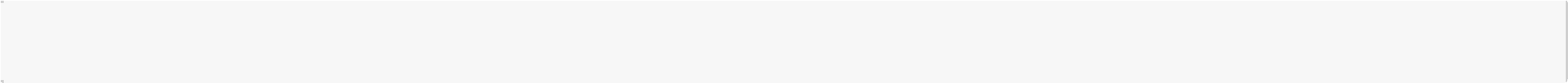
[

'total\_phenols'

,

'color\_intensity'

])



In [9]:

data

.

head

()

[0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

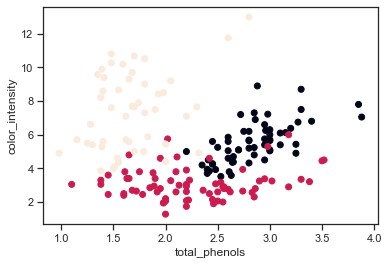
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2

2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2]

[

'class\_0' 'class\_1' 'class\_2'

]



Out[9]:

**total\_phenols**

**color\_intensity**

**0**

2.80

5.64

**1**

2.65

4.38

**2**

2.80

5.68

**3**

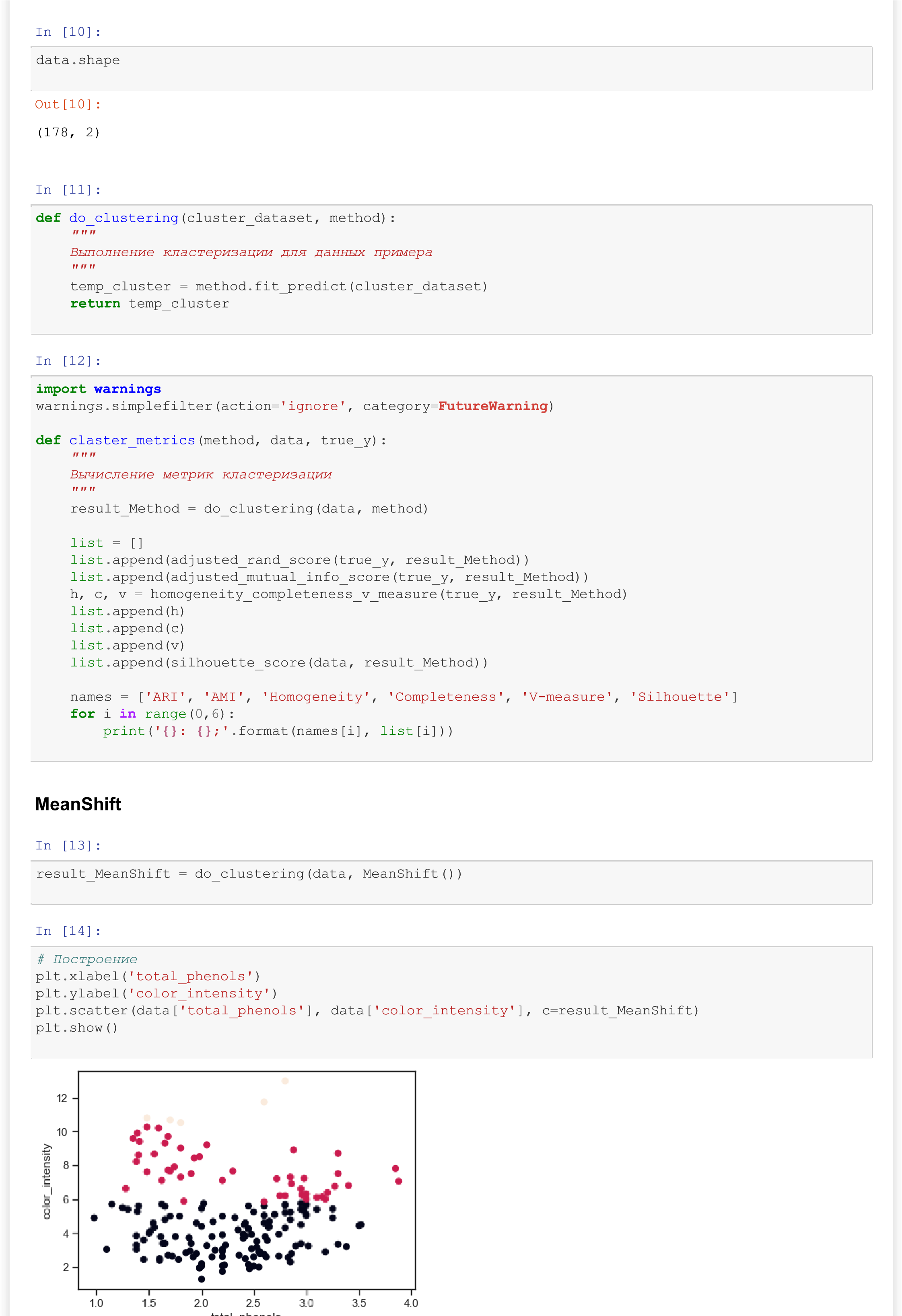
3.85

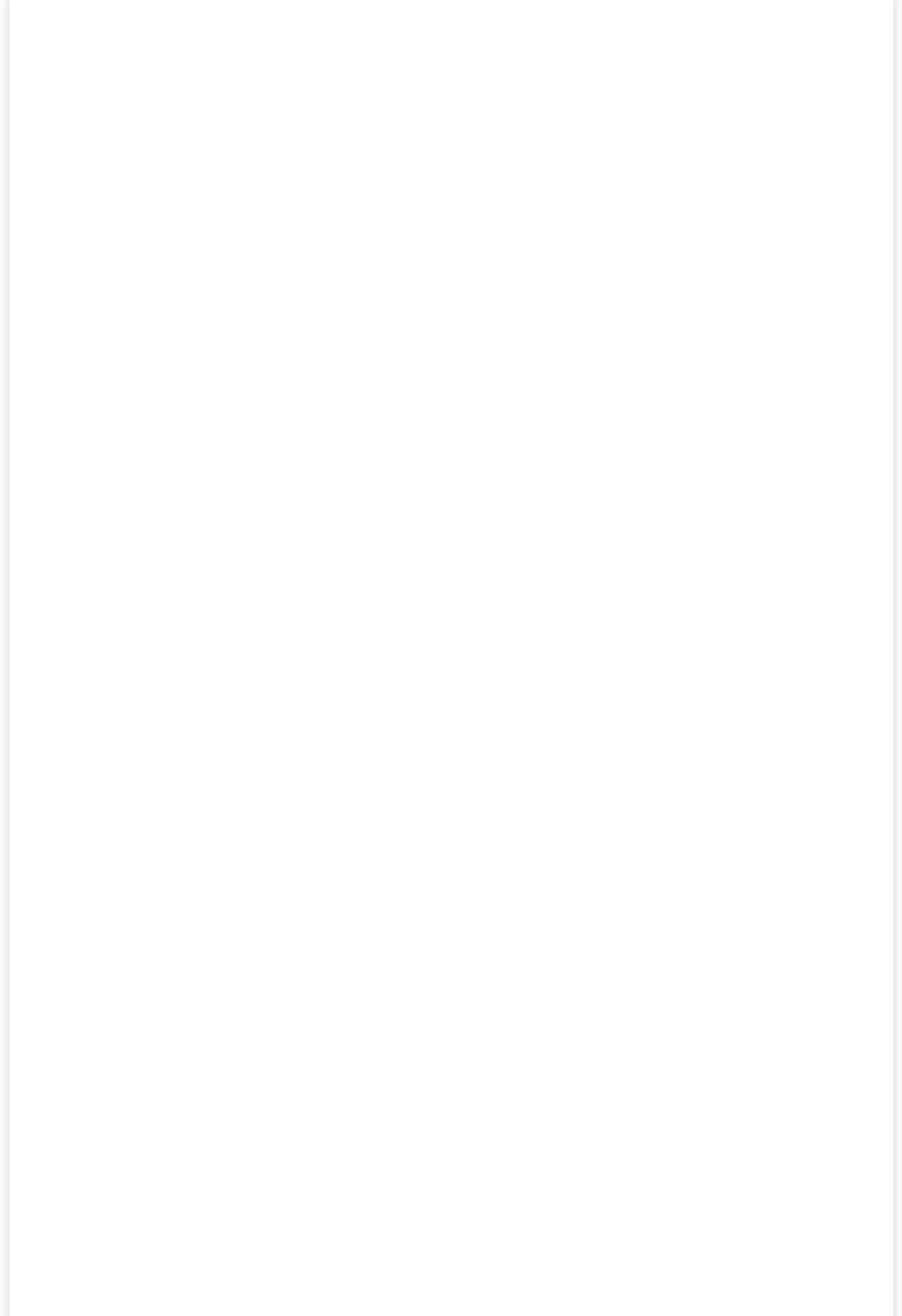
7.80

**4**

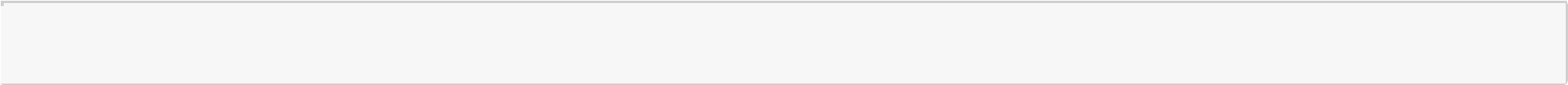
2.80

4.32





**Иерархическая кластеризация**



In [15]:

result\_AgglomerativeClustering

=

do\_clustering

(

data

,

AgglomerativeClustering

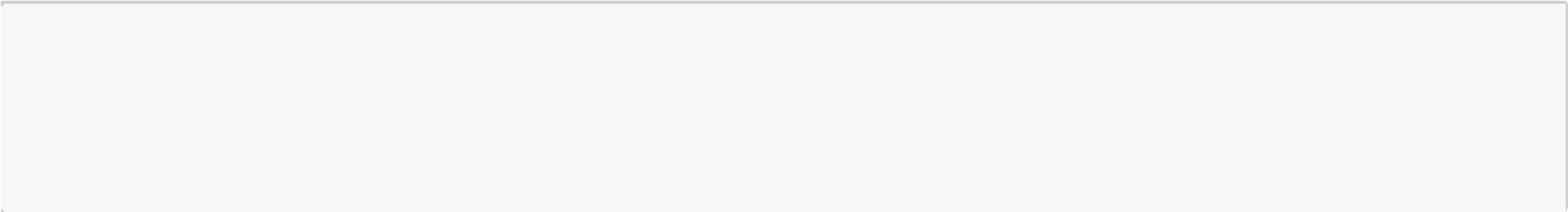
(

n\_clusters

=

3

))



In [16]:

*# Построение*

plt

.

xlabel

(

'total\_phenols'

)

plt

.

ylabel

(

'color\_intensity'

)

plt

.

scatter

(

data

[

'total\_phenols'

]

,

data

[

'color\_intensity'

]

,

c

=

result\_AgglomerativeClustering

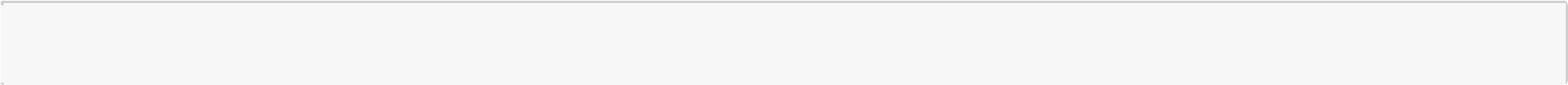
)

plt

.

show

()



In [17]:

result\_AgglomerativeClustering

=

do\_clustering

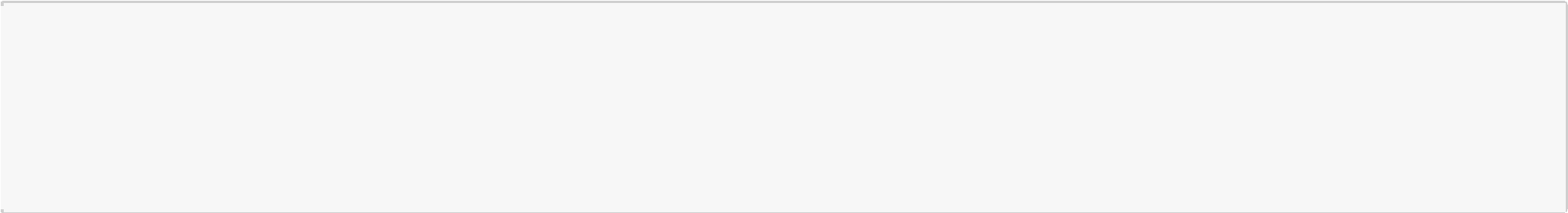
(

data

,

AgglomerativeClustering

())



In [18]:

*# Построение*

plt

.

xlabel

(

'total\_phenols'

)

plt

.

ylabel

(

'color\_intensity'

)

plt

.

scatter

(

data

[

'total\_phenols'

]

,

data

[

'color\_intensity'

]

,

c

=

result\_AgglomerativeClustering

)

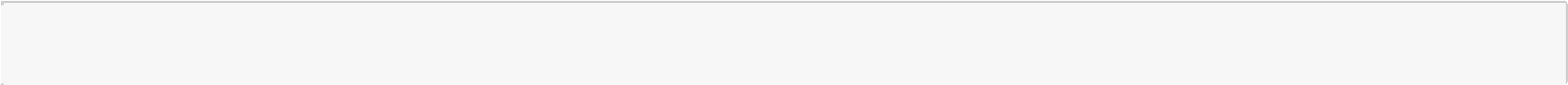
plt

.

show

()

**Сравнение качества кластеризации**



In [19]:

claster\_metrics

(

MeanShift

()

,

data

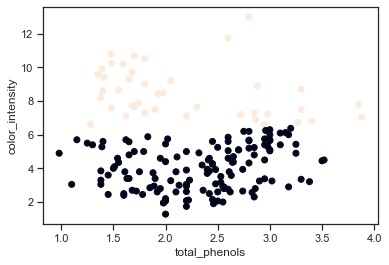
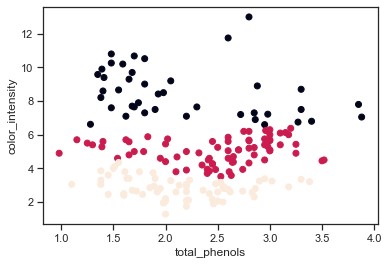
,

wine

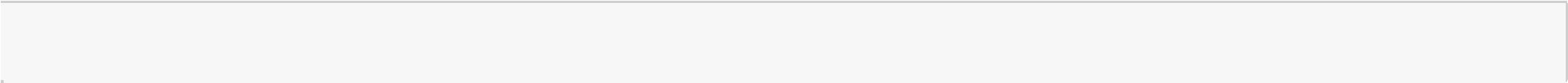
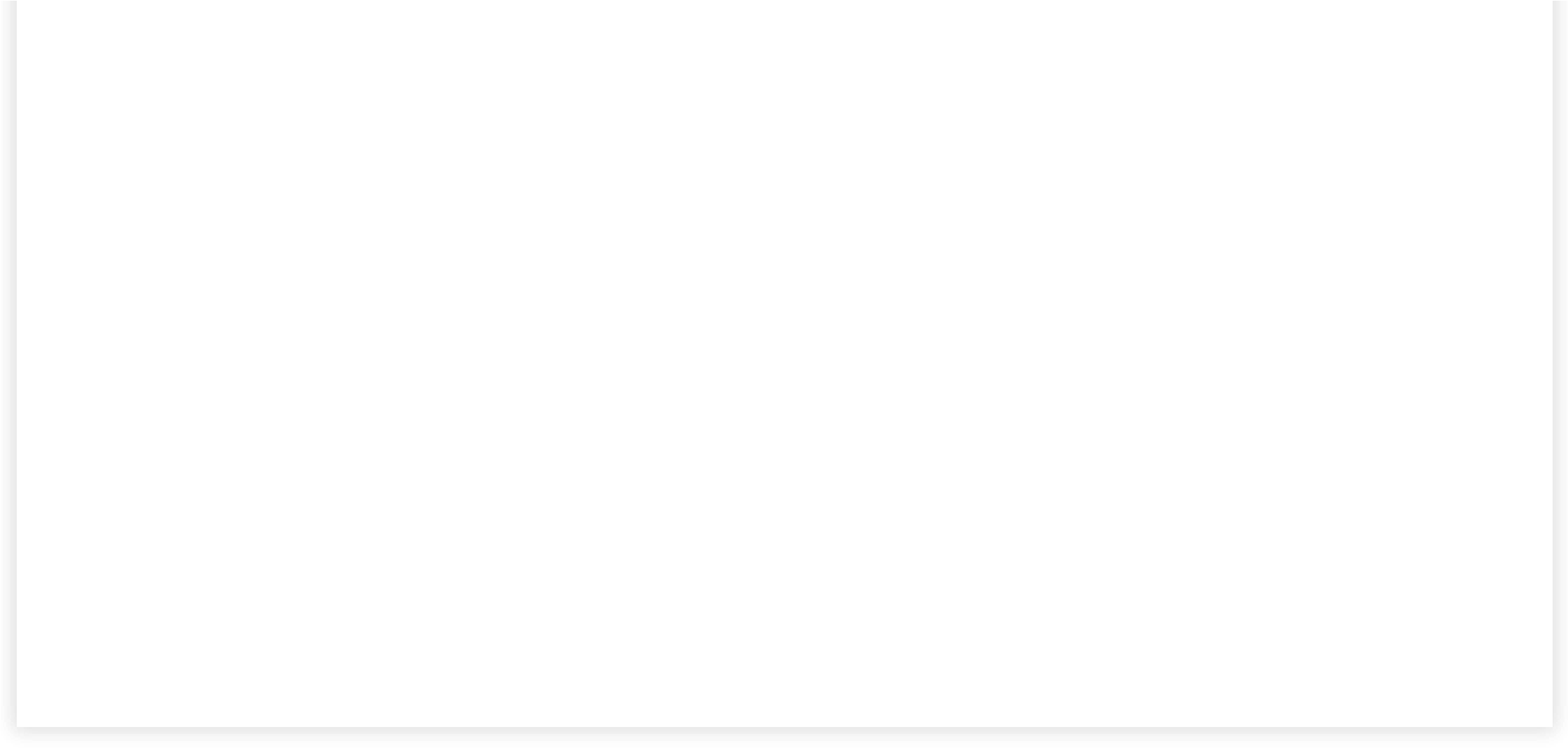
.

target

)



ARI: 0.18489049424576762;



In [20]:

claster\_metrics

(

AgglomerativeClustering

(

n\_clusters

=

3

)

,

data

,

wine

.

target

)

Таким образом, для данного датасета иерархическая кластеризация оказалась более качественной по сравнению с

MeanShift.

ARI: 0.18489049424576762;

AMI: 0.20971343652886495;

Homogeneity: 0.18127363808614247;

Completeness: 0.28264533876567177;

V-measure: 0.22088403968225448;

Silhouette: 0.48025624492483165;

ARI: 0.44829010373340333;

AMI: 0.44489848145061245;

Homogeneity: 0.4475962520065619;

Completeness: 0.4540128430783018;

V-measure: 0.4507817146755018;

Silhouette: 0.4540352632517793;