Рубежный контроль №2¶

Работа Карягин А.Д., группа ИУ5-62Б, вариант 8 (задача №2)

Задача¶

Кластеризуйте данные с помощью двух алгоритмов кластеризации. Алгоритмы для студентов группы ИУ5-62Б: MeanShift и иерархическая кластеризация. Сравните качество кластеризации с помощью следующих метрик качества кластеризации (если это возможно для Вашего набора данных):

1. Adjusted Rand index
2. Adjusted Mutual Information
3. Homogeneity, completeness, V-measure
4. Коэффициент силуэта

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

Набор данных: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_breast_cancer.html#sklearn.datasets.load_breast_cancer

In [55]:

```
import numpy as np
import pandas as pd
from typing import Dict, Tuple
from scipy import stats
from IPython.display import Image
from sklearn import cluster, datasets, mixture
from sklearn.neighbors import kneighbors_graph
from sklearn.preprocessing import StandardScaler
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.cluster import MeanShift, AgglomerativeClustering
from itertools import cycle, islice
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(style="ticks")
from sklearn.datasets import load_breast_cancer
```

In [37]:

```
dataset = load_breast_cancer()
for x in dataset:
    print(x)

data
target
target_names
DESCR
feature_names
filename
```

In [38]:

```
# Признаки
print(dataset.feature_names)
```

```
['mean radius' 'mean texture' 'mean perimeter' 'mean area'
 'mean smoothness' 'mean compactness' 'mean concavity'
 'mean concave points' 'mean symmetry' 'mean fractal dimension'
 'radius error' 'texture error' 'perimeter error' 'area error'
 'smoothness error' 'compactness error' 'concavity error'
 'concave points error' 'symmetry error' 'fractal dimension error'
 'worst radius' 'worst texture' 'worst perimeter' 'worst area'
 'worst smoothness' 'worst compactness' 'worst concavity'
 'worst concave points' 'worst symmetry' 'worst fractal dimension']
```

In [39]:

```
# Метки
print(dataset.target)
```

```
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 1 0 0 0 0 0 0 0 0 0 1 0 1 1 1 1 0 0 1 0 0 1 1 1 1 0 1 0 0 1 1 1 1 0 1 0 0 0
 1 0 1 0 0 1 1 1 0 0 1 0 0 0 1 1 1 0 1 1 0 0 1 1 1 0 0 1 1 1 1 0 1 1 0 1 1 1
 1 1 1 1 1 1 0 0 0 1 0 0 1 1 1 0 0 1 0 1 0 0 1 0 0 1 1 0 1 1 0 1 1 1 1 0 1 1
 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 0 1 0 1 1 0 0 1 1 0 1 1 1 1 0 1 1 0 0 0 1 0 1
 1 0 1 1 1 0 1 1 0 0 1 0 0 0 0 1 0 0 0 1 0 1 1 0 1 1 0 1 0 0 0 0 1 1 0 0 1 1
 1 0 1 1 1 1 1 1 0 0 1 1 0 1 1 0 0 1 0 1 1 1 1 0 1 1 1 1 1 0 1 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 1 0 1 1 0 1 1 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1
 1 0 1 1 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1 0 0 1 1 1 1 1 1
 1 1 0 1 0 1 0 1 1 1 0 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 1 0 0 1 1
 0 1 0 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 1 0 1 1 1 1 1 1
 0 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 0 0 1 0 1 0 1 1 1 1 1 0 1 1 0 1 0 1 0
 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 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 0 0 0 0 0 0 1]
```

In [40]:

```
# Имена меток
print(dataset.target_names)
```

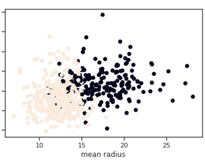
```
['malignant' 'benign']
```

In [41]:

```
# Разделение набора данных
x_axis = dataset.data[:, 0]
y_axis = dataset.data[:, 1]
```

In [42]:

```
# Построение
plt.xlabel(dataset.feature_names[0])
plt.ylabel(dataset.feature_names[1])
plt.scatter(x_axis, y_axis, c=dataset.target)
plt.show()
```



In [43]:

```
data = pd.DataFrame(data= np.c_[dataset.data[:, 0], dataset.data[:, 1]],
                    columns= ['total_phenols', 'color_intensity'])
```

In [44]:

```
data.head()
```

Out[44]:

	total_phenols	color_intensity
0	17.99	10.38
1	20.57	17.77
2	19.69	21.25
3	11.42	20.38
4	20.29	14.34

In [45]:

```
data.shape
```

Out[45]:

```
(569, 2)
```

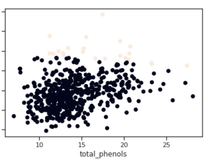
MeanShift¶

In [46]:

```
result_MeanShift = MeanShift().fit_predict(data)
```

In [47]:

```
plt.xlabel('total_phenols')
plt.ylabel('color_intensity')
plt.scatter(data['total_phenols'], data['color_intensity'], c=result_MeanShift)
plt.show()
```



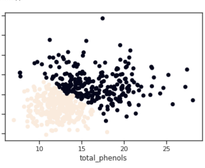
Иерархическая кластеризация¶

In [48]:

```
result_AgglomerativeClustering = AgglomerativeClustering(n_clusters=2).fit_predict(data)
```

In [49]:

```
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 [50]:

```
def do_clustering(cluster_dataset, method):
    """
    Выполнение кластеризации для данных примера
    """
    temp_cluster = method.fit_predict(cluster_dataset)
    return temp_cluster
```

In [51]:

```
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)

def claster_metrics(method, data, true_y):
    """
    Вычисление метрик кластеризации
    """
    result_Method = do_clustering(data, method)

    list = []
    list.append(adjusted_rand_score(true_y, result_Method))
    list.append(adjusted_mutual_info_score(true_y, result_Method))
    h, c, v = homogeneity_completeness_v_measure(true_y, result_Method)
    list.append(h)
    list.append(c)
    list.append(v)
    list.append(silhouette_score(data, result_Method))

    names = ['ARI', 'AMI', 'Homogeneity', 'Completeness', 'V-measure', 'Silhouette']
    for i in range(0,6):
        print('{}: {};'.format(names[i], list[i]))
```

In [52]:

```
claster_metrics(MeanShift(), data, dataset.target)
```

```
ARI: 0.02597499009392261;
AMI: 0.017014445921275596;
Homogeneity: 0.011959534897521055;
Completeness: 0.04824827952634769;
V-measure: 0.019167843518729348;
Silhouette: 0.4651482752156396;
```

In [53]:

```
claster_metrics(AgglomerativeClustering(n_clusters=2), data, dataset.target)
```

```
ARI: 0.4807058606925514;
AMI: 0.3936323965676893;
Homogeneity: 0.4026221401760883;
Completeness: 0.3865554161475689;
V-measure: 0.3944252283891643;
Silhouette: 0.4127921205305984;
```

Иерархическая кластеризация оказалась более качественной по сравнению с MeanShift.

In [ ]: