Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления» Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Технологии машинного обучения»

Отчет по лабораторной работе №6 «Методы обучения без учителей»

Выполнил:
студент группы ИУ5-64Б
Низовцев Р.А.

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю.Е.

Москва, 2022 г.

Лабораторная работа №6

Низовцев Роман

Импорт библиотек

```
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.manifold import TSNE
from sklearn.cluster import KMeans, MiniBatchKMeans
from sklearn.cluster import AgglomerativeClustering
from sklearn.cluster import AffinityPropagation
from sklearn.cluster import MeanShift, SpectralClustering
from sklearn.cluster import DBSCAN
from sklearn.mixture import GaussianMixture
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 import datasets, ensemble
from sklearn.datasets import *
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline
sns.set(style='ticks')
```

Загрузка данных

```
data : pd.DataFrame = pd.read_csv('/wine.csv', index_col=0)
d1 = data.drop(columns=['Cultivars'])
```

Анализ данных

```
# Первые пять строк датасета
d1.head()
```

	Alcohol	Malic acid	Ash	Alcalinity of ash	Magnesium	Total Phenols	Flavanoids	Nonflavan phen
0	14.23	1.71	2.43	15.6	127	2.80	3.06	
1	13.20	1.78	2.14	11.2	100	2.65	2.76	
2	13.16	2.36	2.67	18.6	101	2.80	3.24	
3	14.37	1.95	2.50	16.8	113	3.85	3.49	
4	13.24	2.59	2.87	21.0	118	2.80	2.69	

```
# Последние пять строк датасета
d1.tail()
```

	Alcohol	Malic acid	Ash	Alcalinity of ash	Magnesium	Total Phenols	Flavanoids	Nonflav ph
173	13.71	5.65	2.45	20.5	95	1.68	0.61	
174	13.40	3.91	2.48	23.0	102	1.80	0.75	
175	13.27	4.28	2.26	20.0	120	1.59	0.69	
176	13.17	2.59	2.37	20.0	120	1.65	0.68	
177	14.13	4.10	2.74	24.5	96	2.05	0.76	

```
# Размер датасета
d1.shape
```

```
(178, 13)
```

```
# Типы столбцов
d1.dtypes
```

```
Alcohol                         float64
Malic acid                      float64
Ash                             float64
Alcalinity of ash               float64
Magnesium                         int64
Total Phenols                   float64
Flavanoids                      float64
Nonflavanoid phenols            float64
Proanthocyanins                 float64
Color intensity                 float64
Hue                             float64
OD280/OD315 of diluted wines    float64
Proline                           int64
dtype: object
```

```
# Пустые значения
d1.isnull().sum()
```

```
Alcohol                         0
Malic acid                      0
Ash                             0
Alcalinity of ash               0
Magnesium                       0
Total Phenols                   0
Flavanoids                      0
Nonflavanoid phenols            0
Proanthocyanins                 0
Color intensity                 0
Hue                             0
OD280/OD315 of diluted wines    0
Proline                         0
dtype: int64
```

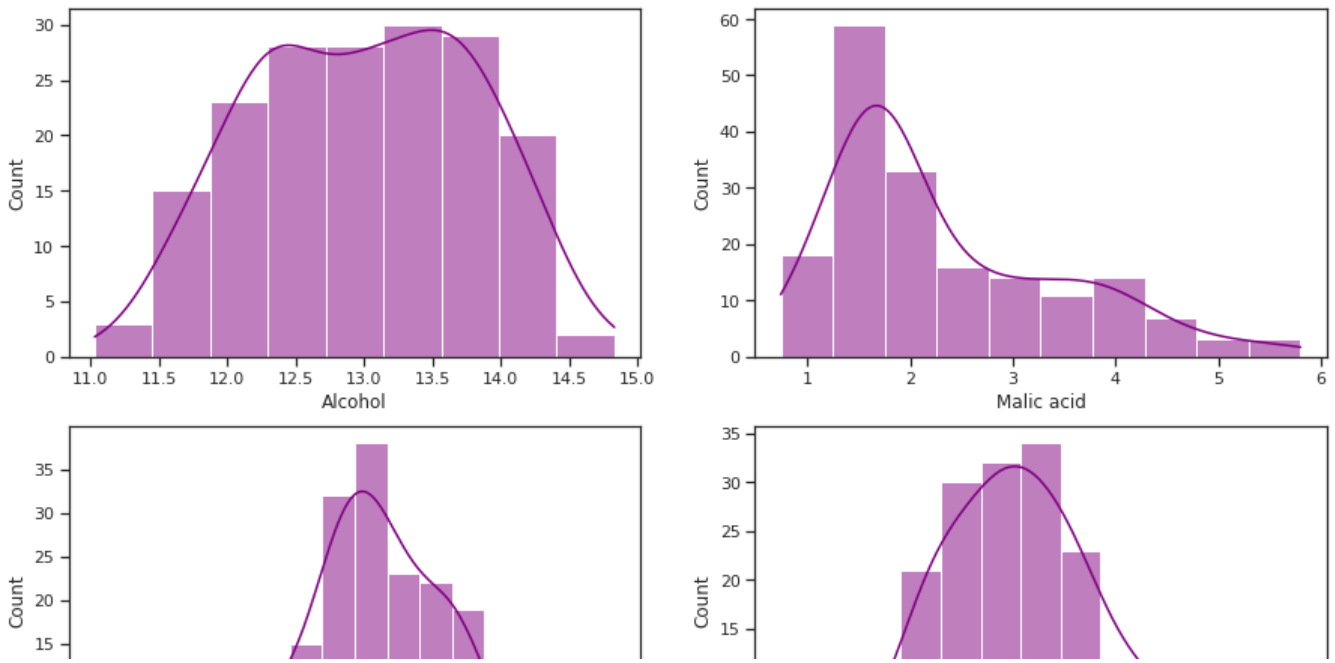
```
# Дублирующиеся значения
d1.duplicated().sum()
```

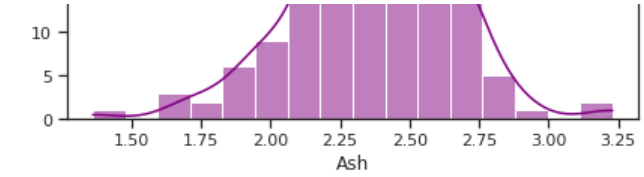
```
0
```

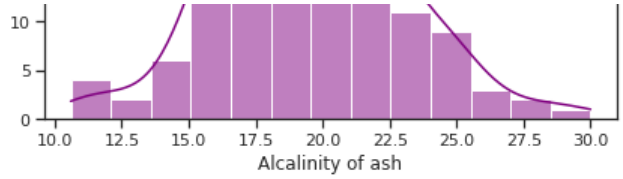
```
# Статистические характеристки датасета
d1.describe()
```

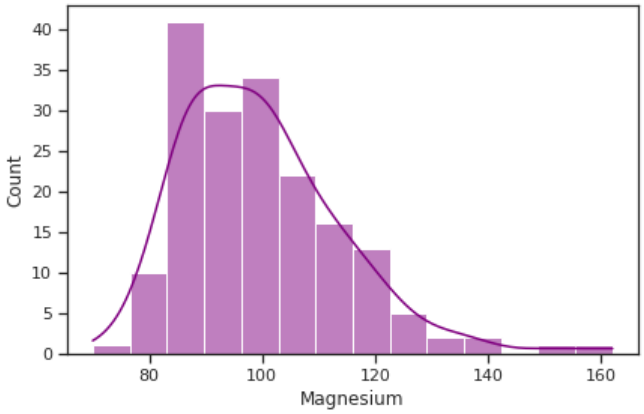
	Alcohol	Malic acid	Ash	Alcalinity of ash	Magnesium	Total Phenols	Flav
count	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178
mean	13.000618	2.336348	2.366517	19.494944	99.741573	2.295112	2
std	0.811827	1.117146	0.274344	3.339564	14.282484	0.625851	0
min	11.030000	0.740000	1.360000	10.600000	70.000000	0.980000	0
25%	12.362500	1.602500	2.210000	17.200000	88.000000	1.742500	1
50%	13.050000	1.865000	2.360000	19.500000	98.000000	2.355000	2
75%	13.677500	3.082500	2.557500	21.500000	107.000000	2.800000	2
max	14.830000	5.800000	3.230000	30.000000	162.000000	3.880000	5

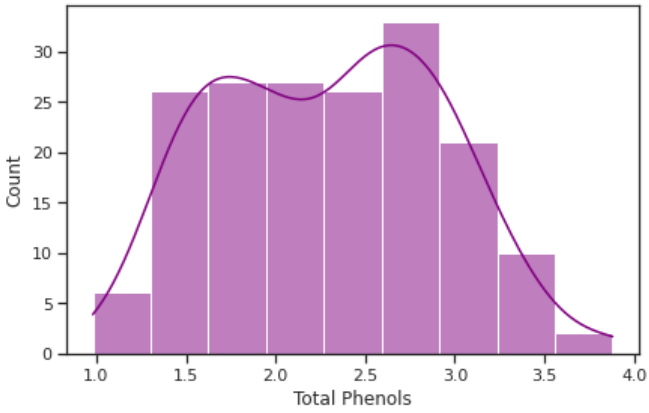
```
# Гистограммы столбцов
fig, ax = plt.subplots(7, 2, figsize=(15, 35))
for i, column in zip(np.arange(0, ax.size), d1.columns):
    sns.histplot(data=d1, x=column, color='purple', kde=True, ax=ax[int(i/2), i%
fig.delaxes(ax[6, 1])
plt.show()
```

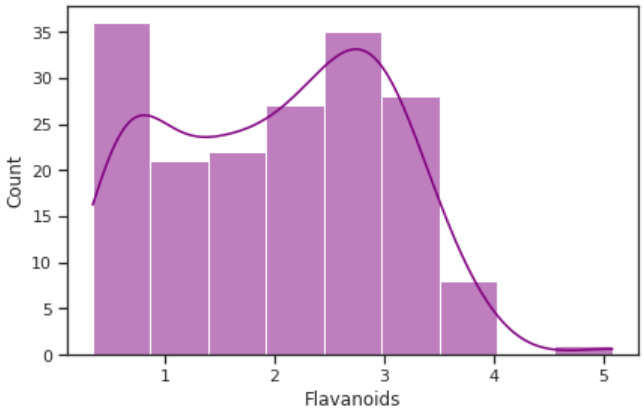


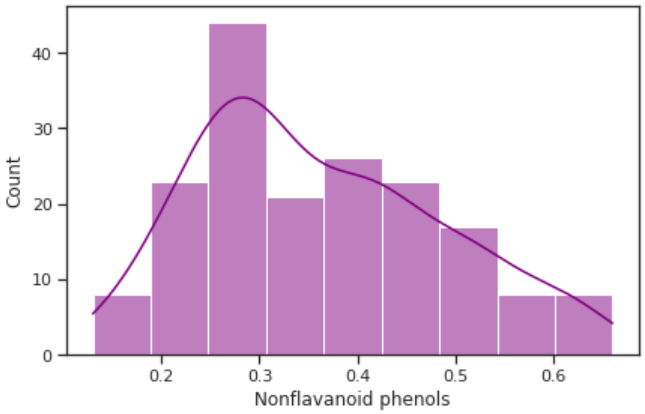


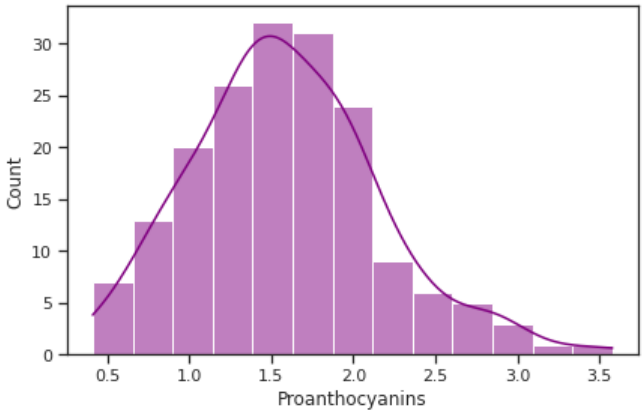


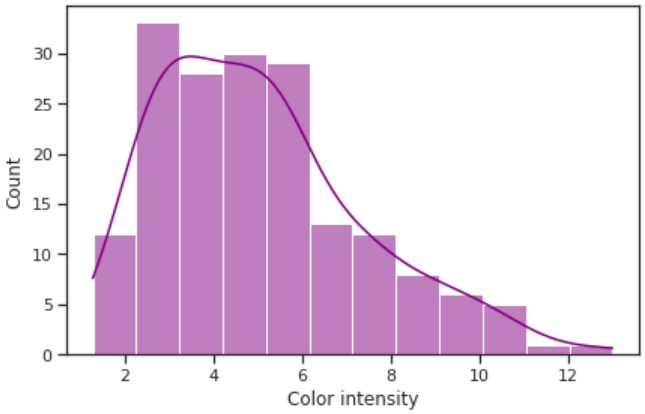












Масштабирование признаков

```
# Масштабирование данных
scaler = StandardScaler()

d1_scaled = pd.DataFrame(scaler.fit_transform(X=d1), columns=d1.columns)
```

Снижение размерности

Метод главных компонент

[ ] ↳ *Скрыто 2 ячейки.*

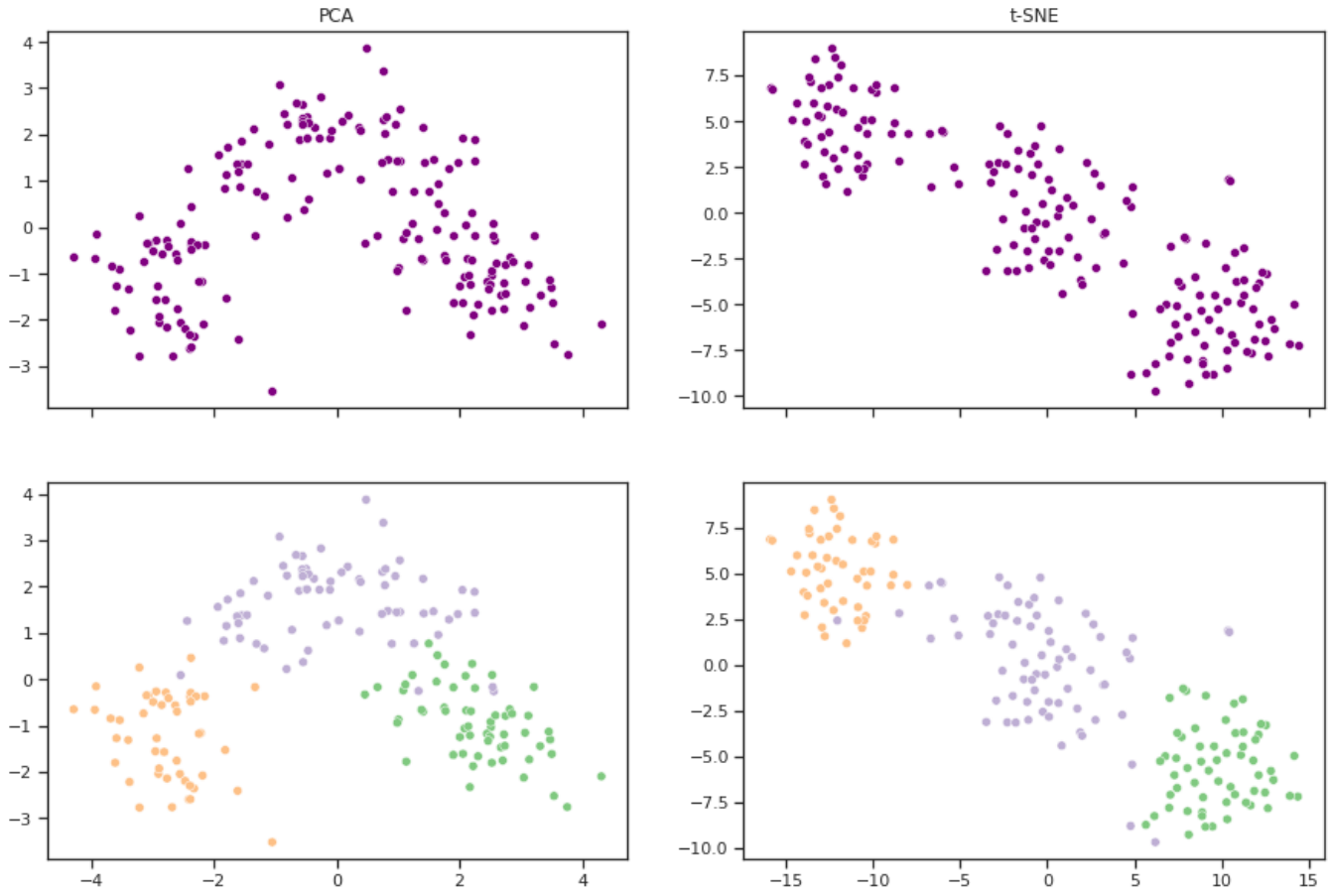
Алгоритм t-SNE

[ ] ↳ *Скрыта 1 ячейка.*

Визуализация новых датасетов

```
# Метки классов

class_labels = data['Cultivars']
```

```
# Визуалиация датасетов
fig, ax = plt.subplots(2, 2, sharex='col', figsize=(15, 10))
sns.scatterplot(data=d2, x='PC1', y='PC2', color='purple', legend=False, ax=ax[0
sns.scatterplot(data=d3, x='C1', y='C2', color='purple', legend=False, ax=ax[0,
sns.scatterplot(data=d2, x='PC1', y='PC2', hue=class_labels, palette='Accent', l
sns.scatterplot(data=d3, x='C1', y='C2', hue=class_labels, palette='Accent', leg
ax[0, 0].set(xlabel=None, ylabel=None, title='PCA')
ax[0, 1].set(xlabel=None, ylabel=None, title='t-SNE')
ax[1, 0].set(xlabel=None, ylabel=None)
ax[1, 1].set(xlabel=None, ylabel=None)
plt.show()
```



В графике датасета D3 (t-SNE) кластеры выявлены более явно, чем в D2 (PCA). Но если учитывать метки классов, то ситуация становится обратной: класстеры более выражены у датасета D2.

Кластеризация

```
# Списки датасетов, моделей и оценок
dataset_list = [d1_scaled, d2, d3]

method_list = []
d1_scores_list = []
d2_scores_list = []
d3_scores_list = []


def do_clustering(datasets, method):
    """
    Фунция кластеризации нескольких датасетов

    :param datasets: Датасеты
    :param method: Метод кластеризации
    :return: Список кластеров, центры кластеров
    """
    results = []
    for dataset in datasets:
        result = method.fit_predict(dataset)
        results.append(result)

    return results
```

```
def do_clustering_with_centers(datasets, method):
    """
    Фунция кластеризации нескольких датасетов с центрами кластеризации

    :param datasets: Датасеты
    :param method: Метод кластеризации
    :return: Список кластеров, центры кластеров
    """
    results = []
    centers = []
    for dataset in datasets:
        result = method.fit_predict(dataset)
        centers.append(method.cluster_centers_)
        results.append(result)

    return results, centers
```

```
def cluster_metrics(datasets, clusters, cluster_true):
    """
    Функция оценки качества кластеризации различных наборов данных

    :param datasets: Датасеты
    :param clusters: Кластеры
    :param cluster_true: Истинное значение класстеров
    :return: Таблицу с метриками для каждого датасета
    """

    datasets_names = ['D1', 'D2', 'D3']

    ari = []
    ami = []
    hl = []
    cl = []
    vl = []
    sl = []

    for cluster, dataset in zip(clusters, datasets):
        ari.append(adjusted_rand_score(cluster_true, cluster))
        ami.append(adjusted_mutual_info_score(cluster_true, cluster))

        h, c, v = homogeneity_completeness_v_measure(cluster_true, cluster)
        hl.append(h)
        cl.append(c)
        vl.append(v)

        try:
            sil_score = silhouette_score(dataset, cluster)
        except:
            sil_score =None
        sl.append(sil_score)

    result = pd.DataFrame({'Datasets':datasets_names,
                           'ARI':ari, 'AMI':ami,
                           'Homogeneity':hl,
                           'Completeness':cl,
                           'V-measure':vl, 'Silhouette':sl})
    return result
```

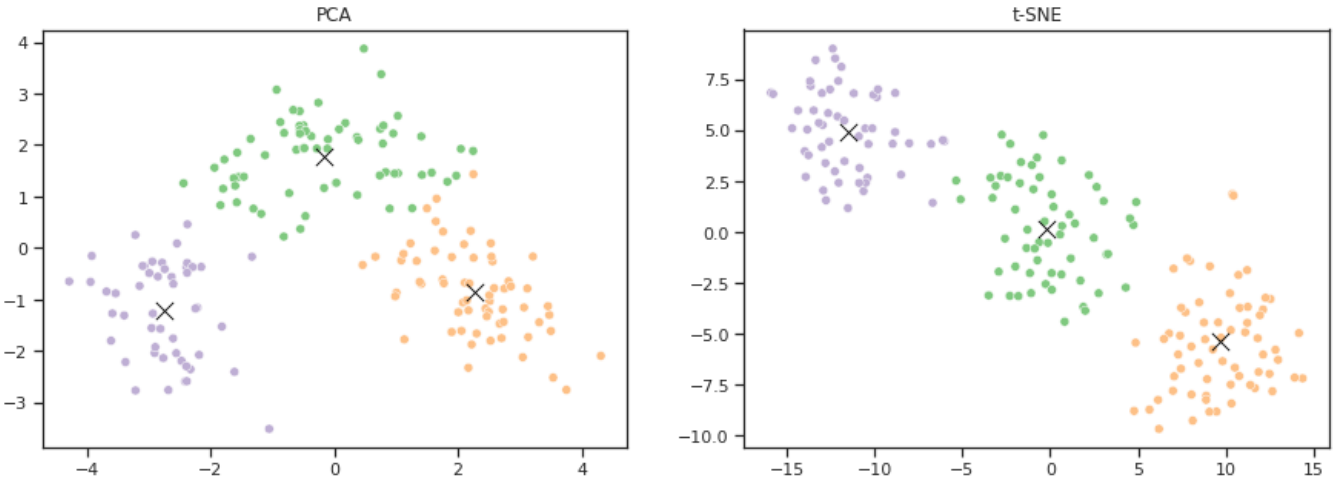
```
def plot_clusters(clusters):
    """
    Фунция визуализации кластеров

    :param clusters: Кластеры для визуализации
    """
    f, a = plt.subplots(1, 2, figsize=(15, 5))
    sns.scatterplot(data=d2, x='PC1', y='PC2', hue=clusters[0], palette='Accent'
    sns.scatterplot(data=d3, x='C1', y='C2', hue=clusters[1], palette='Accent',
    a[0].set(xlabel=None, ylabel=None, title='PCA')
    a[1].set(xlabel=None, ylabel=None, title='t-SNE')
    plt.show()


def plot_clusters_with_centers(clusters, clusters_centers):
    """
    Фунция визуализации кластеров

    :param clusters: Кластеры для визуализации
    :param clusters_centers: Центры кластеров
    """
    f, a = plt.subplots(1, 2, figsize=(15, 5))
    sns.scatterplot(data=d2, x='PC1', y='PC2', hue=clusters[0], palette='Accent'
    sns.scatterplot(data=clusters_centers[0], x=clusters_centers[0][:, 0], y=clu
                    ax=a[0], marker='x', s=128, color='black')
    sns.scatterplot(data=d3, x='C1', y='C2', hue=clusters[1], palette='Accent',
    sns.scatterplot(data=clusters_centers[1], x=clusters_centers[1][:, 0], y=clu
                    ax=a[1], marker='x', s=128, color='black')
    a[0].set(xlabel=None, ylabel=None, title='PCA')
    a[1].set(xlabel=None, ylabel=None, title='t-SNE')
    plt.show()
```

Метод K-средних

```
%%time
# Применение метода
k_means = KMeans(n_clusters=3, random_state=4)
type(k_means)
k_means_clusters, k_means_centers = do_clustering_with_centers(dataset_list, k_m
```

```
CPU times: user 131 ms, sys: 2.12 ms, total: 133 ms
Wall time: 76.6 ms
```

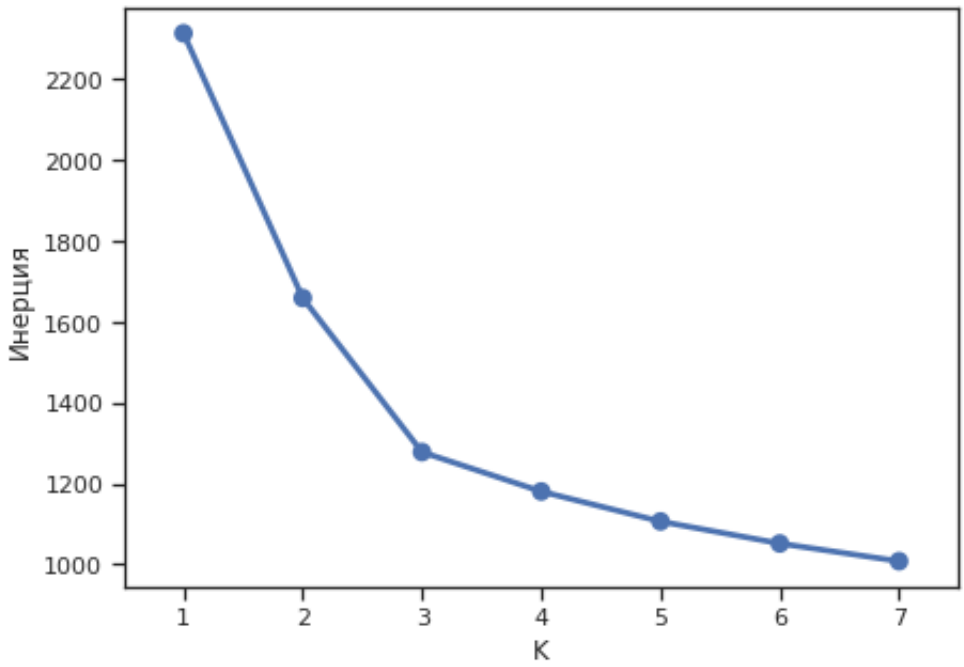
```
# Визуализация кластеров
plot_clusters_with_centers(k_means_clusters[1:], k_means_centers[1:])
```




```
# Оценка метода
cluster_metrics(dataset_list, k_means_clusters, class_labels)
```

	Datasets	ARI	AMI	Homogeneity	Completeness	V-measure	Silhouette
0	D1	0.897495	0.874579	0.878843	0.872964	0.875894	0.284859
1	D2	0.895058	0.880815	0.883981	0.880158	0.882065	0.561051
2	D3	0.802459	0.793332	0.799049	0.791971	0.795494	0.614013

```
def plot_inertia(dataset, method):
    """
    Функция для визуализации инерции
    :param dataset: Датасет
    :param method: Метод кластеризации
    """
    _, a = plt.subplots(figsize=(7, 5))
    inertia = []
    temp_x = dataset
    for k in range(1, 8):
        kmeans = method(n_clusters=k, random_state=4).fit(temp_x)
        inertia.append(kmeans.inertia_)
    sns.pointplot(x=np.arange(1, 8), y=inertia, ax=a)
    plt.xlabel('K')
    plt.ylabel('Инерция')
    plt.show()


# График инерции для датасета D1
plot_inertia(d1_scaled, KMeans)
```




```
# Занесение результатов в соответствующие списки
method_list.append('K-Means')
d1_scores_list.append(adjusted_rand_score(class_labels, k_means_clusters[0]))
d2_scores_list.append(adjusted_rand_score(class_labels, k_means_clusters[1]))
d3_scores_list.append(adjusted_rand_score(class_labels, k_means_clusters[2]))
```

Affinity Propagation

[ ] ↳ *Скрыто 4 ячейки.*

Spectral clustering

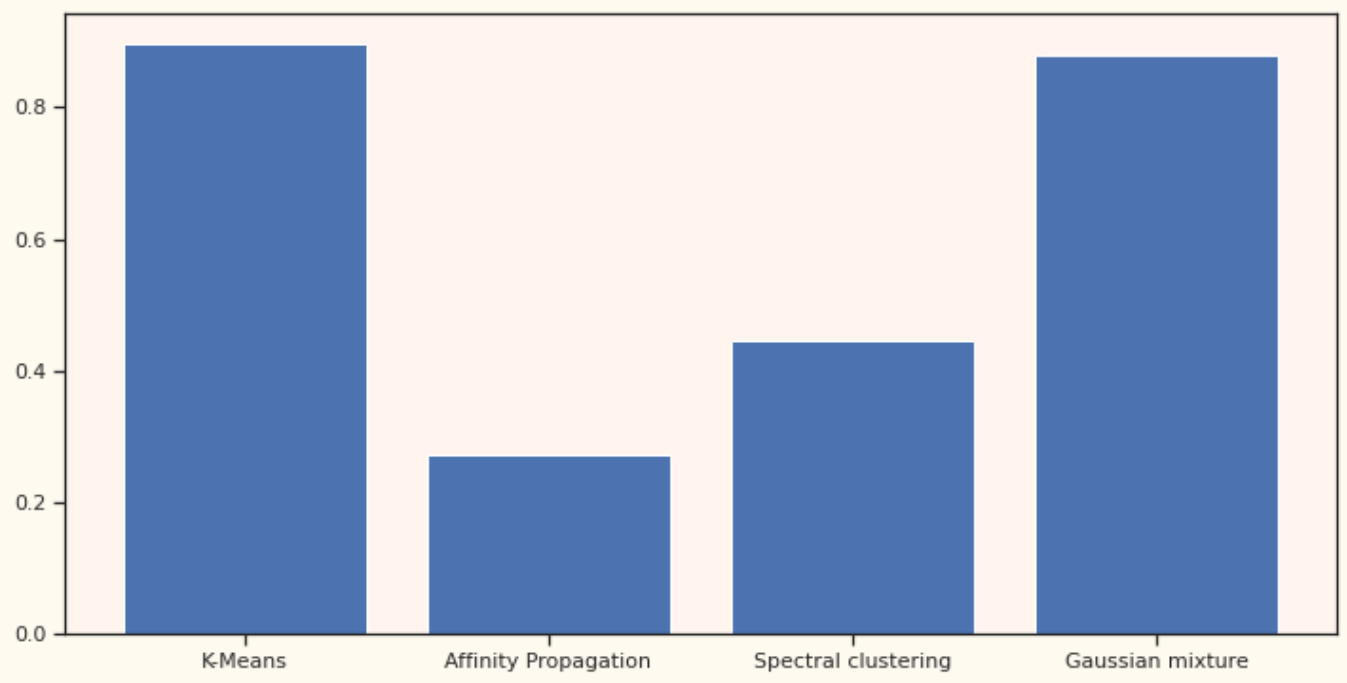
[ ] ↳ *Скрыто 4 ячейки.*

Gaussian mixture

[ ] ↳ *Скрыто 4 ячейки.*

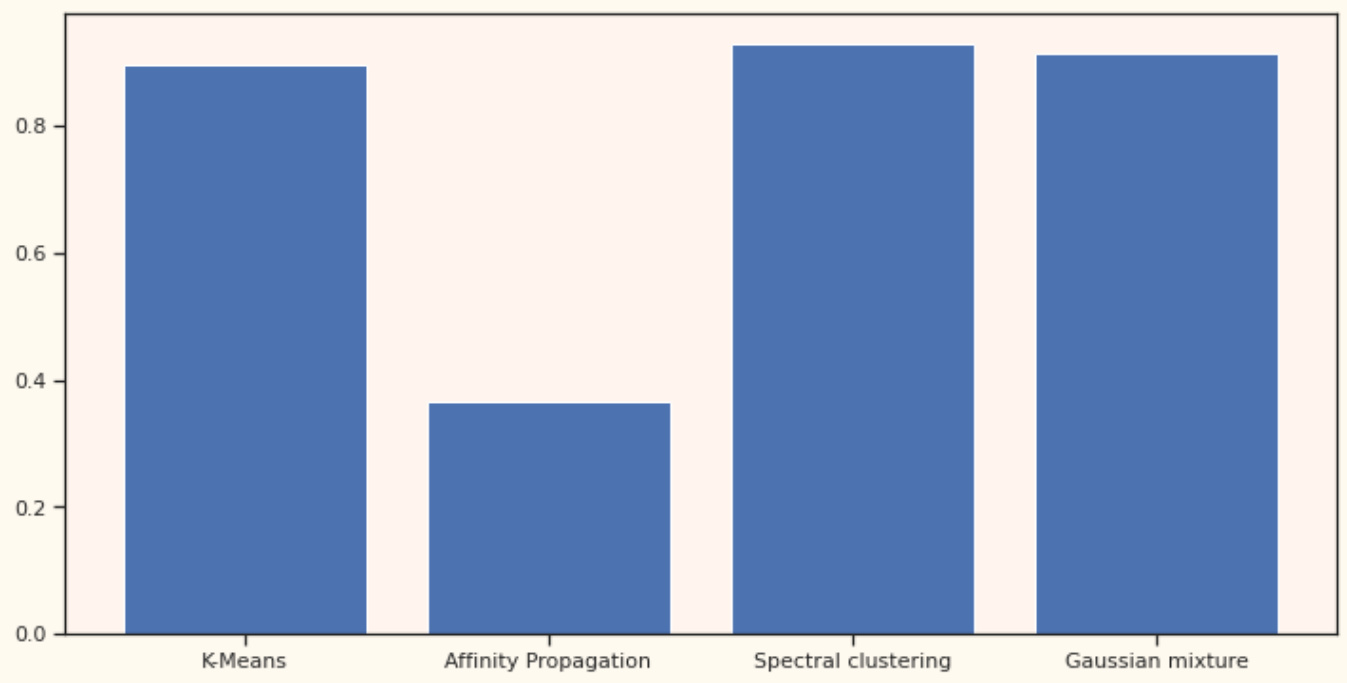
Оценка качества методов кластеризации

```
# Оценка качества кластеризации для датасета D1
x = method_list
y = d1_scores_list

fig, ax = plt.subplots()

ax.bar(x, y)

ax.set_facecolor('seashell')
fig.set_facecolor('floralwhite')
fig.set_figwidth(12)    #  ширина Figure
fig.set_figheight(6)    #  высота Figure

plt.show()
```



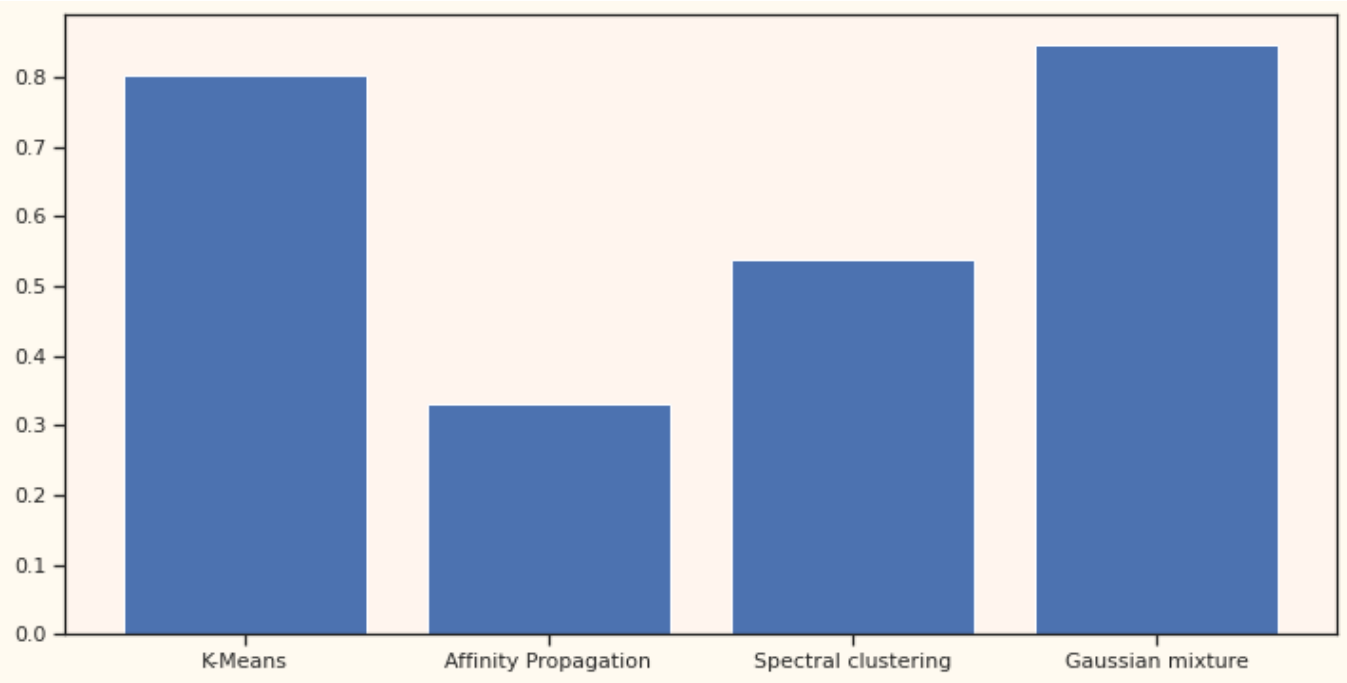
Для датасета D1 лучшим методом оказался метод k-means

```
# Оценка качества кластеризации для датасета D2
x = method_list
y = d2_scores_list

fig, ax = plt.subplots()

ax.bar(x, y)

ax.set_facecolor('seashell')
fig.set_facecolor('floralwhite')
fig.set_figwidth(12)    #  ширина Figure
fig.set_figheight(6)    #  высота Figure

plt.show()
```



Для датасета D2 лучшим методом также оказался метод Spectral clustering

```
# Оценка качества кластеризации для датасета D3

x = method_list
y = d3_scores_list

fig, ax = plt.subplots()

ax.bar(x, y)

ax.set_facecolor('seashell')
fig.set_facecolor('floralwhite')
fig.set_figwidth(12)    #  ширина Figure
fig.set_figheight(6)    #  высота Figure

plt.show()
```



Для датасета D3 лучшим методом оказалась Gaussian mixture