

ГУАП

КАФЕДРА № 41

ОТЧЕТ  
ЗАЩИЩЕН С ОЦЕНКОЙ  
ПРЕПОДАВАТЕЛЬ

Старший преподаватель  
должность, уч. степень, звание

подпись, дата

В.В. Боженко  
инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №4

КЛАСТЕРИЗАЦИЯ 2024

по курсу: ВВЕДЕНИЕ В АНАЛИЗ ДАННЫХ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. № 4217

подпись, дата

Д.М. Никитин  
инициалы, фамилия

Санкт-Петербург 2024

1. **Цель работы:** изучение алгоритмов и методов кластеризации на практике.

2. **Вариант и задание:**

4 вариант

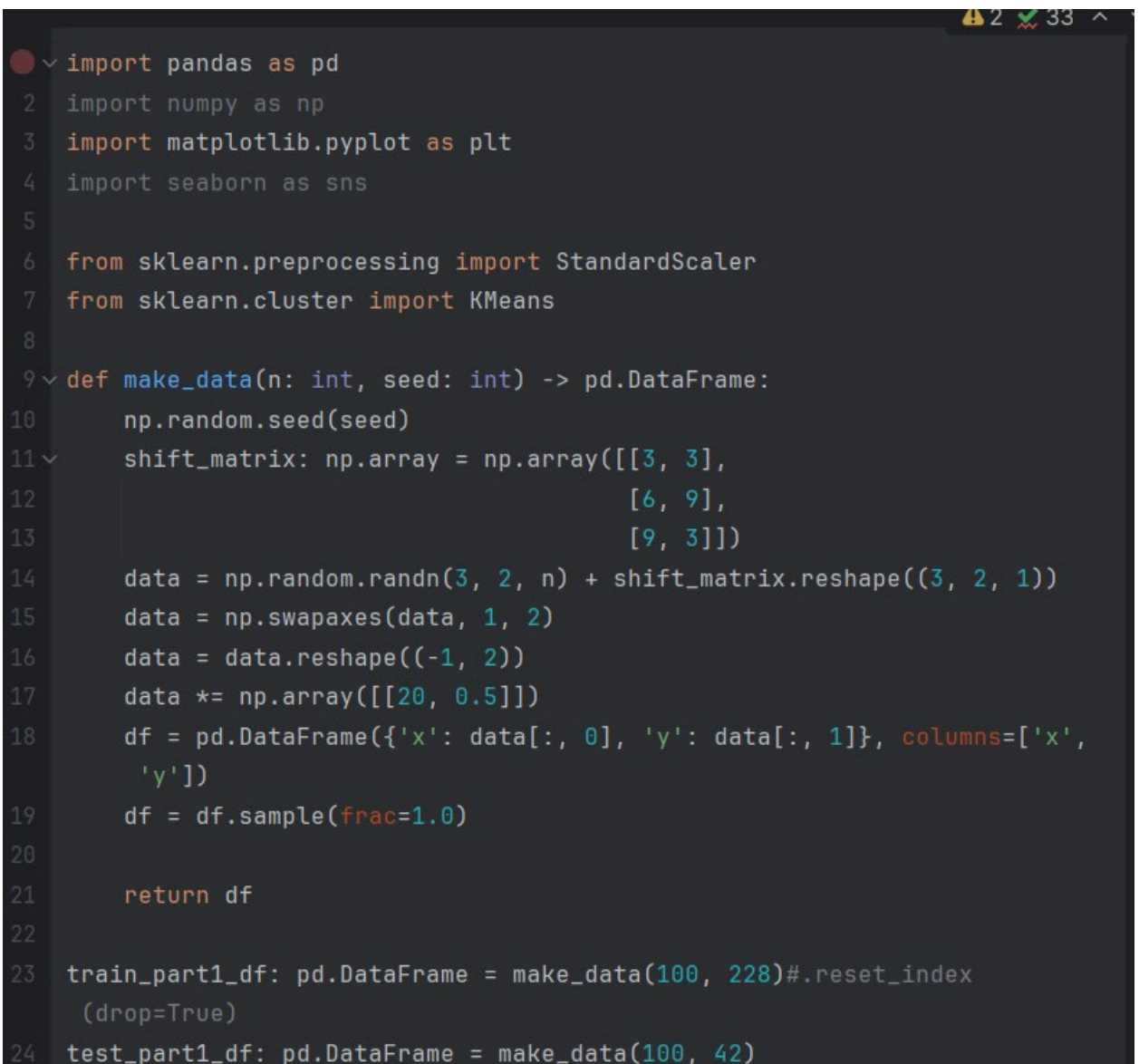
Набор данных 4heart2.csv

Данные о болезнях сердца:

1. возраст: возраст пациента (лет)
2. анемия: снижение количества эритроцитов или гемоглобина (логическое значение)
3. высокое кровяное давление: если у пациента гипертония (логическое значение)
4. креатининфосфокиназа (КФК): уровень фермента КФК в крови (мкг/л)
5. диабет: если у пациента диабет (логическое значение)
6. фракция выброса: процент крови, покидающей сердце при каждом сокращении (в процентах)
7. тромбоциты: тромбоциты в крови (килотромбоциты/ мл)
8. пол: женщина или мужчина (бинарный)
9. креатинин сыворотки: уровень креатинина сыворотки в крови (мг/дл)
10. натрий сыворотки: уровень натрия сыворотки в крови (мэкв/л)
11. курение: если пациент курит или нет (логическое)
12. время: период наблюдения (дни)
13. событие смерти: если пациент умер в течение периода наблюдения (логическое значение)

3. **Ход работы:**

Сначала используется функция для создания набора данных. Она показана на рисунке 1.



```

1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import seaborn as sns
5
6 from sklearn.preprocessing import StandardScaler
7 from sklearn.cluster import KMeans
8
9 def make_data(n: int, seed: int) -> pd.DataFrame:
10     np.random.seed(seed)
11     shift_matrix: np.array = np.array([[3, 3],
12                                         [6, 9],
13                                         [9, 3]])
14     data = np.random.randn(3, 2, n) + shift_matrix.reshape((3, 2, 1))
15     data = np.swapaxes(data, 1, 2)
16     data = data.reshape((-1, 2))
17     data *= np.array([[20, 0.5]])
18     df = pd.DataFrame({'x': data[:, 0], 'y': data[:, 1]}, columns=['x',
19                                                                    'y'])
19     df = df.sample(frac=1.0)
20
21     return df
22
23 train_part1_df: pd.DataFrame = make_data(100, 228).reset_index
24                                     (drop=True)
25 test_part1_df: pd.DataFrame = make_data(100, 42)

```

Рисунок 1 – Функция для создания набора данных

Набор данных создан, далее проводится стандартизация данных. В результате применения получается DataFrame , содержащий заданное число объектов в каждой группе (всего 3 группы) с двумя признаками: 'x' и 'y'. Код на рисунке 2.

```
1 scaler = StandardScaler()
2
3 scaled_test_part1_df: pd.DataFrame = pd.DataFrame(scaler.fit_transform
  (test_part1_df), columns=test_part1_df.columns)
4 scaled_train_part1_df: pd.DataFrame = pd.DataFrame(scaler.fit_transform,
  (train_part1_df), columns=train_part1_df.columns)
5
6 print(train_part1_df.shape)
7 test_part1_df
✓ [277] 16ms
(300, 2)
```

	$\bar{x}$	$\bar{y}$
79	20.248622	2.860085
12	64.839245	1.530115
204	170.998691	1.174679
137	123.729086	4.339307
99	55.308257	0.928515
47	81.142445	0.839772
205	192.456999	1.256437
15	48.754249	1.650774
42	57.687034	0.696258
190	115.837555	4.050793

Рисунок 2 – Скейл данных

Стандартизация данных проведена. При этом стандартизировано как значение  $x$ , так и  $y$ . При кластеризации они оба стандартизируются. Далее проведём кластеризацию данных. См. рис. 3, 4.

```

1 kmeans = KMeans(n_clusters=3, random_state=0)
2 train_clusters_n3 = kmeans.fit_predict(scaled_train_part1_df)
3
4 # Получение центров кластеров
5 cluster_centers = kmeans.cluster_centers_
6
7 def show_cluster_scatter(scaled_df: pd.DataFrame, clusters: pd.Series,
8   cluster_centers_funk: pd.DataFrame, title: str) -> None:
9
10     # Покраска объектов из разных кластеров разными цветами
11     plt.scatter(scaled_df.iloc[:, 0], scaled_df.iloc[:, 1], c=clusters,
12       cmap='viridis', marker='o', edgecolor='k', s=50)
13
14     # Пометка центров кластеров
15     plt.scatter(cluster_centers_funk[:, 0], cluster_centers_funk[:, 1],
16       c='red', marker='X', s=200, edgecolor='k', label='Центры')
17
18     plt.legend()
19
20     plt.xlabel(scaled_df.columns[0])
21     plt.ylabel(scaled_df.columns[1])
22
23     plt.title(title)
24
25     plt.show()
26
27 show_cluster_scatter(scaled_train_part1_df, train_clusters_n3,
28   cluster_centers, '3-means Кластеризация (тренировочная)')

```

Рисунок 3 – Реализация кластеризации

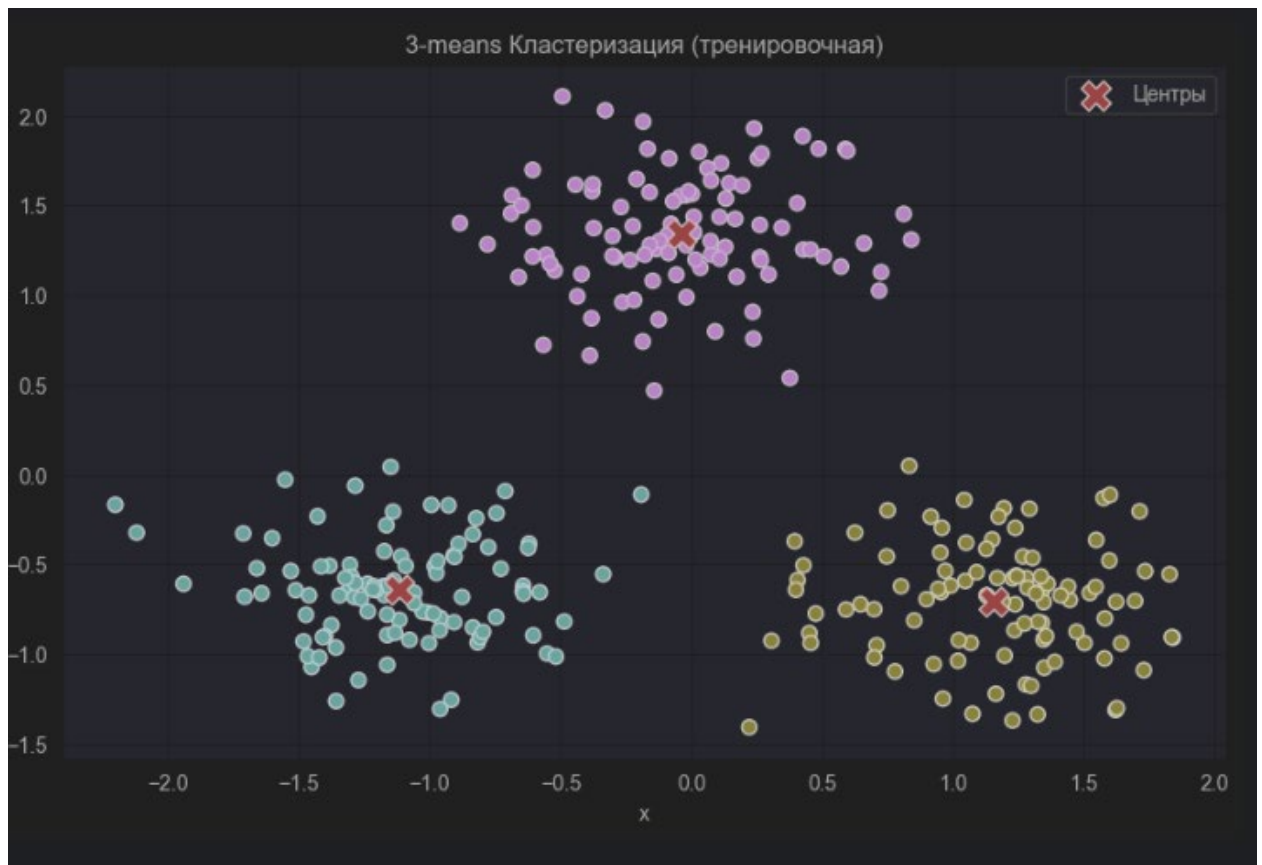


Рисунок 4 – Графическая кластеризация

Было проведено обучение и кластеризация тренировочных данных. На графике чётко видно, что имеется 3 кластера. См. рис. 5.

```

1 clusters_test_n3 = kmeans.predict(scaled_test_part1_df)
2
3 cluster_centers = kmeans.cluster_centers_
4
5 show_cluster_scatter(scaled_test_part1_df, clusters_test_n3,
   cluster_centers, '3-means Кластеризация на тестовых данных')
✓ [279] 199ms

```

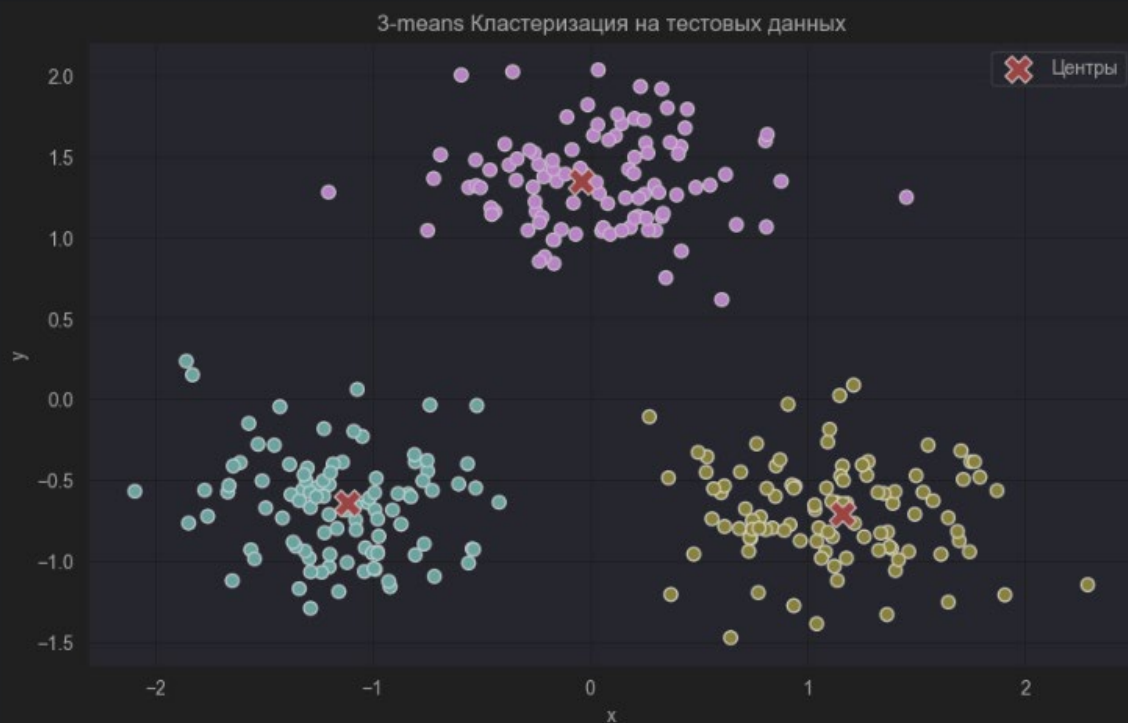


Рисунок 5 – Кластеризация на тестовых данных

Далее была проведена кластеризация на тестовых данных. Модель с кластеризацией справляется хорошо. Снова на графике отчётливо видно 3 кластера данных, раскрашенных разным цветом.

Далее будет проведена кластеризация данных для кластеризации при  $n\_clusters=2$  и  $n\_clusters=4$  и проведено сравнение.

Сначала будет проведена кластеризация данных при  $n\_clusters=2$ . См. рис.6, 7.

```
1 kmeans = KMeans(n_clusters=2, random_state=0)
2 train_clusters_n2 = kmeans.fit_predict(scaled_train_part1_df)
3 |
4 cluster_centers = kmeans.cluster_centers_
5 show_cluster_scatter(scaled_train_part1_df, train_clusters_n2,
6                       cluster_centers, '2-means Кластеризация (тренировочная)')
7
8 clusters_test_n2 = kmeans.predict(scaled_test_part1_df)
9
10 cluster_centers = kmeans.cluster_centers_
11
12 show_cluster_scatter(scaled_test_part1_df, clusters_test_n2,
13                       cluster_centers, '2-means Кластеризация на тестовых данных')
✓ [280] 433ms
```

Рисунок 6 – Реализация 2 кластеризации



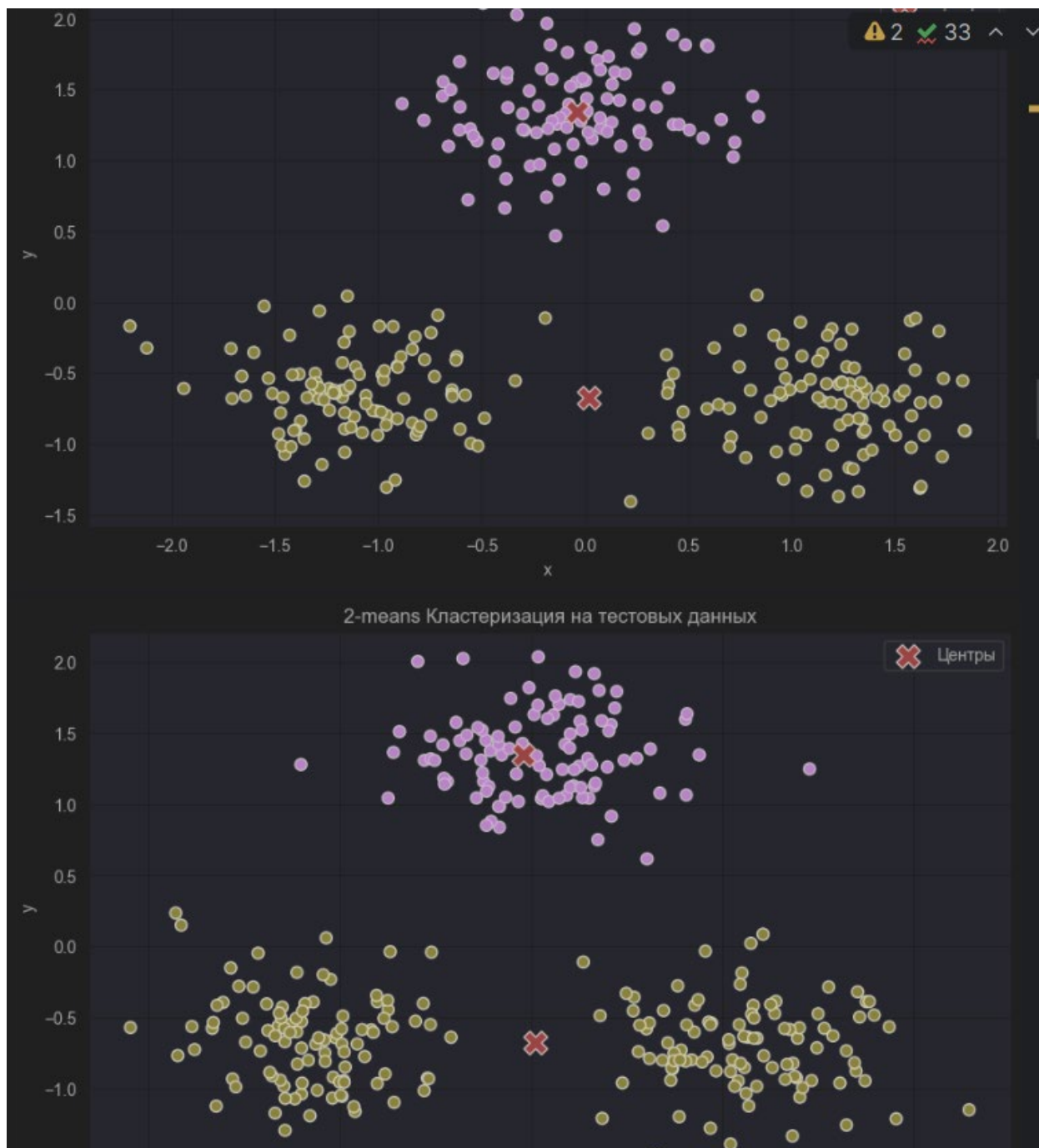


Рисунок 7 – Результат 2 кластеризации

Была проведена кластеризация данных при  $n\_clusters=2$ . Далее будет проведена кластеризация данных при  $n\_clusters=4$ . При  $n\_clusters=2$  модель становится явно менее точной, так как фактическое количество кластеров на графике больше. Однако модель не может сделать больше кластеров, чем было выбрано. Поэтому она располагает центры так, чтобы минимизировать сумму квадратов расстояний между точками данных и ближайшим центром. Следовательно, между двумя нижними облаками сходства больше, чем у

каждого из них с верхним. См. рис. 8, 9.

```
1 kmeans = KMeans(n_clusters=4, random_state=0)
2 train_clusters_n4 = kmeans.fit_predict(scaled_train_part1_df)
3
4 cluster_centers = kmeans.cluster_centers_
5 show_cluster_scatter(scaled_train_part1_df, train_clusters_n4,
6                       cluster_centers, '4-means Кластеризация (тренировочная)')
7
8 clusters_test_n4 = kmeans.predict(scaled_test_part1_df)
9
10 cluster_centers = kmeans.cluster_centers_
11
12 show_cluster_scatter(scaled_test_part1_df, clusters_test_n4,
13                       cluster_centers, '4-means Кластеризация на тестовых данных')
✓ [281] 431ms
```

Рисунок 8 – Реализация 4 кластеризации



Рисунок 9 – Результат 4 кластеризации

Кластеризация  $n\_clusters=4$  делит нижнее левое облако ещё на 2 части. Это означает, что в нём данные менее однородны, чем в остальных облаках. Оно содержит больше точек и имеет слегка вытянутую форму. Алгоритм старается "равномерно" распределить центры кластеров, что приводит к появлению двух центров (желтого и голубого) в этой области. Это может быть признаком того, что данные имеют слабую естественную границу между

этими подгруппами, либо это искусственное разделение, вызванное выбором фиксированного числа кластеров. Таким образом мы можем узнать, какие кластеры различаются сильнее.

В нашем случае самым лучшим вариантом будет разделение на 3 кластера, так как фактически на графике имеется 3 кластера, нужно лишь разграничить их и найти центры. Таким образом: 2 кластера слишком мало, так как снизу визуально имеется 2 группы, 4 кластера слишком много, так как левая группа выглядит естественным единым кластером. См. рис. 10, 11.

```
1 # Список для хранения инерции
2 inertia = []
3
4 # Число кластеров
5 range_clusters = range(1, 11)
6
7 # Расчет инерции для каждого k
8 for k in range_clusters:
9     kmeans = KMeans(n_clusters=k, random_state=42)
10    kmeans.fit(scaled_train_part1_df)
11    inertia.append(kmeans.inertia_)
12
13 # Построение графика
14 plt.figure(figsize=(8, 5))
15 plt.plot(range_clusters, inertia, marker='o')
16 plt.title('Метод локтя')
17 plt.xlabel('Число кластеров (k)')
18 plt.ylabel('Инерция')
19 plt.grid(True)
20 plt.show()
✓ [282] 255ms
```

Рисунок 10 – Реализация инерции

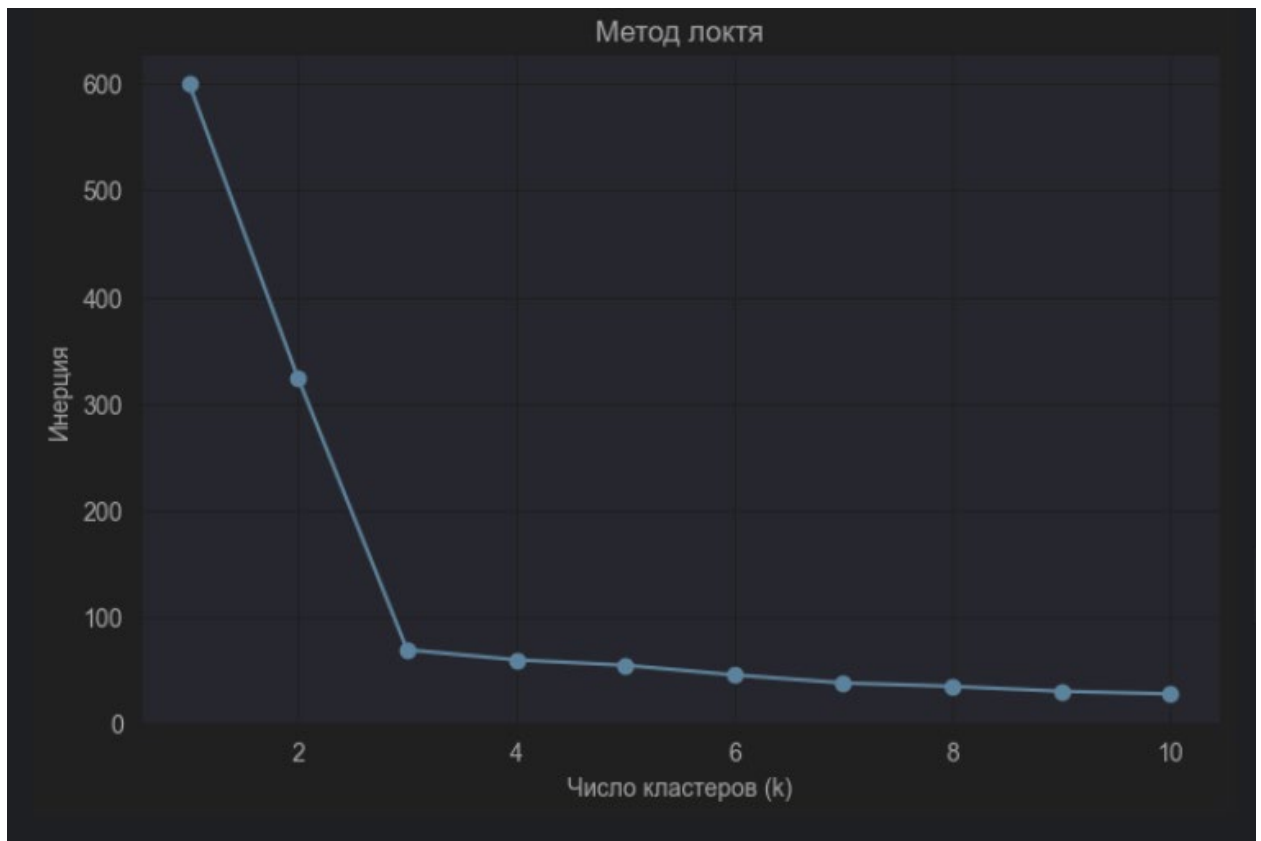


Рисунок 11 - Результат

На графике изображена инерция, это сумма квадратов расстояний от точек до их центров кластеров. Чем больше кластеров, тем меньше инерция.

На графике ищем точку, где снижение инерции начинает замедляться ("локоть"). Это и есть оптимальное число кластеров  $k$ . В нашем случае оптимальное число кластеров равно 3, так как излом находится на точке 3. Это также подтверждается и визуальной информацией графика.

Далее будет проведена кластеризация для набора данных, предложенного вариантом 4.

Сначала будут открыты и проведена предварительная обработка данных. См. рис. 12.

```

1 df = pd.read_csv("4heart2.csv", sep=",")
2 df.isnull().any()
✓ [283] 12ms

```

		Length: 13, dtype: bool		
			10 01	<unnamed>
age		False		
anaemia		False		
creatinine_phosphokinase		False		
diabetes		False		
ejection_fraction		False		
high_blood_pressure		False		
platelets		False		
serum_creatinine		False		
serum_sodium		False		
sex		False		

Рисунок 12 – Нормализация данных

Так как набор данных является тестовым, дубликатов и пропусков в нём нет. Данные чистые. Далее будет сделан вывод матрицы диаграмм рассеяния. См. рис. 13, 14.

```

1 # Выделение целевой переменной
2 target = df['DEATH_EVENT']
3
4 # Исключение целевой переменной из данных
5 features = df.drop(columns=['DEATH_EVENT'])
6
7 # Построение матрицы диаграмм рассеяния
8 # sns.pairplot(df, hue='DEATH_EVENT', palette='Set1', diag_kind='kde')
9 # plt.suptitle('Матрица диаграмм рассеяния с окраской по DEATH_EVENT',
10 #             y=1.02)
10 # plt.show()
✓ [285] 10ms

```

Рисунок 13 – Выделение целевой переменной



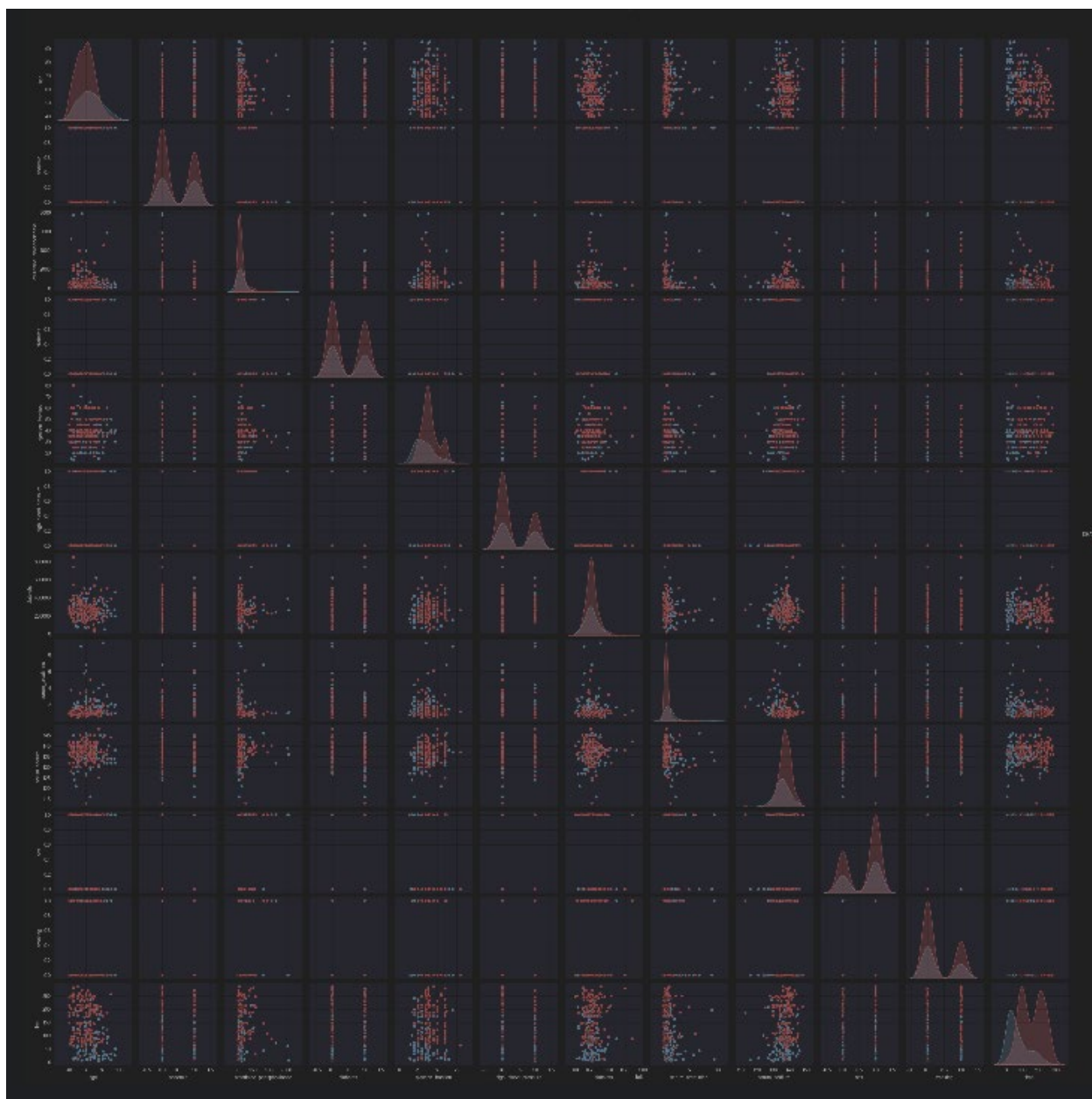


Рисунок 14 – Часть полученной матрицы

На данной матрице диаграмм рассеяния можно увидеть зависимость летального исхода от различных факторов. Красным цветом изображены летальные случаи, а синим - не летальные. Учесть все факторы одним взглядом на матрицу проблематично. Например, можно заметить, что чем больше период наблюдения за пациентом, тем больше смертность. Возможно это из-за того, что время наблюдения увеличивается по мере серьёзности и излечимости заболевания. Далее будет произведён скейл данных. См. рис. 15.

```
1 # Определяем числовые столбцы, исключая бинарные
2 numerical_columns = features.select_dtypes(include=['float64',
3 'int64']).columns
4 binary_columns = ['anaemia', 'diabetes', 'high_blood_pressure', 'sex',
5 'smoking']
6 columns_to_scale = [col for col in numerical_columns if col not in
7 binary_columns]
8
9 # Стандартизация только числовых непрерывных данных
10 scaler = StandardScaler()
11 scaled_features = scaler.fit_transform(features[columns_to_scale])
12
13 # Собираем итоговый DataFrame
14 scaled_df = features.copy()
15 scaled_df[columns_to_scale] = scaled_features
16
17 scaled_df
18 ✓ [301] 33ms
```

	age	anaemia	creatinine_phosphokinase	diabetes
0	1.192945	0	0.000166	0
1	-0.491279	0	7.514640	0
2	0.350833	0	-0.449939	0
3	-0.912335	1	-0.486071	0
4	0.350833	1	-0.435486	1
5	2.456114	1	-0.552141	0

Рисунок 15 - Реализация

Производится скейл данных, но только числовых значений, не изменяя bool значения. Далее будет выполнена кластеризация методом k-means. См. рис. 16, 17.



```

1 # Вычисление метрики "инерции" для разных значений k
2 inertia = []
3 k_range = range(1, 11)
4
5 for k in k_range:
6     kmeans = KMeans(n_clusters=k, random_state=42)
7     kmeans.fit(features)
8     inertia.append(kmeans.inertia_)
9
10 # Построение графика метода локтя
11 plt.figure(figsize=(8, 5))
12 plt.plot(k_range, inertia, 'bo-', markersize=8)
13 plt.xlabel('Количество кластеров (k)', fontsize=12)
14 plt.ylabel('Инерция', fontsize=12)
15 plt.title('Метод локтя для определения оптимального k', fontsize=14)
16 plt.grid()
17 plt.show()

```

✓ [302] 283ms

Рисунок 16 – Реализация кластеризации и метода локтя

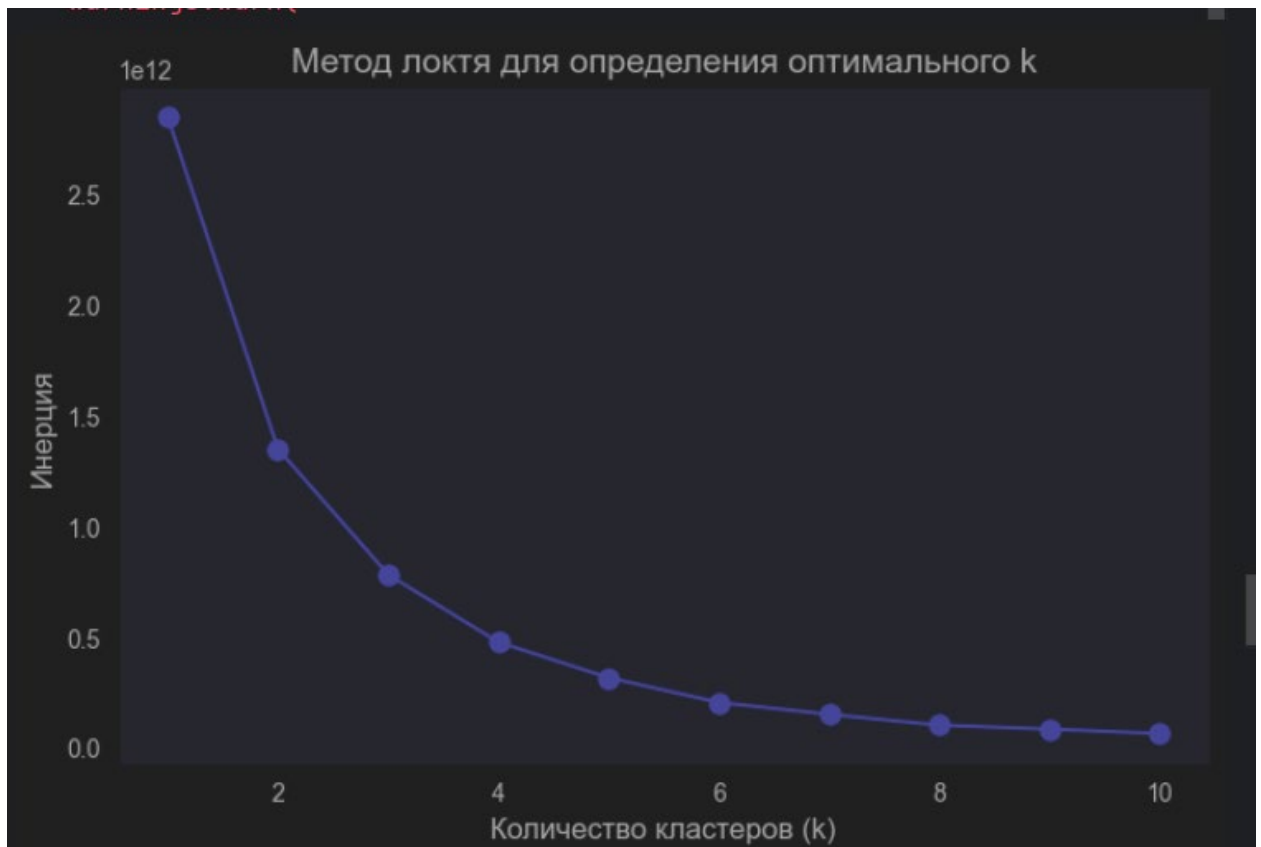


Рисунок 17 – График метода локтя

Было выбрано 4 кластера, так как по методу локтя это является

оптимальным значением. См. рис. 18.

```
1 optimal_k = 4
2 kmeans = KMeans(n_clusters=optimal_k, random_state=0)
3 clusters = kmeans.fit_predict(scaled_df[columns_to_scale])
4
5 scaled_df['Cluster'] = clusters
6
7 cluster_means = scaled_df.groupby('Cluster').mean()
8
9 print("Средние значения по каждому кластеру:")
10 cluster_means
```

✓ [306] 36ms

Средние значения по каждому кластеру:

A:\applications\MiniConda\Lib\site-packages\sklearn\cluster\\_kmeans.py:1429: UserWarning: KMeans is known to have a memory leak on Windows with MKL, when there are less chunks than available threads. You can avoid it by setting the environment variable OMP\_NUM\_THREADS=2.

warnings.warn()

Cluster	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine
0	-0.293135	0.176471	3.229052	0.529412	-0.111629	0.235294	0.351005	0.056
1	-0.125232	0.491935	-0.228666	0.395161	0.120563	0.427419	0.063636	-0.256
2	0.942066	0.479167	-0.225919	0.375000	-0.112874	0.395833	-0.243887	1.185
3	-0.224610	0.381818	-0.142683	0.445455	-0.069401	0.263636	-0.019558	-0.237

Рисунок 18 – Вывод средних значений групп

Кластер 0 включает людей с возрастом ниже среднего, с высокой частотой диабета и анемии, а также с умеренно повышенным уровнем креатинкиназы. У этих людей фракция выброса крови близка к норме, но несколько понижена. Также среди этого кластера наблюдается умеренное распространение гипертонии. Уровень тромбоцитов в пределах нормы, а креатинин в сыворотке и уровень натрия находятся в пределах нормы. Большинство людей в кластере — мужчины, а доля курящих людей небольшая. Время наблюдения в среднем близко к минимальному, что может указывать на относительно быструю выписку.

Кластер 1 включает людей с возрастом, близким к среднему по набору данных, с более выраженной частотой анемии и с низким уровнем креатинкиназы. У этих людей наблюдается умеренная частота диабета, а фракция выброса крови выше средней. Гипертония встречается в этом кластере часто, что может указывать на проблемы с артериальным давлением. Уровень тромбоцитов и креатинина в сыворотке ниже среднего, а натрия в сыворотке — в пределах нормы. Большинство людей в этом кластере — мужчины, курящих людей больше, чем в кластере 0. Время наблюдения у этого кластера относительно низкое, что может свидетельствовать о более позднем периоде выписки.

Кластер 2 включает людей с выше среднего возраста, с частотой анемии,

близкой к средней, и с низким уровнем креатинкиназы. Среди этих людей наблюдается низкая частота диабета. Фракция выброса крови в этом кластере также близка к среднему значению. Гипертония встречается достаточно часто, и уровень тромбоцитов в среднем ниже нормы. Креатинин в сыворотке значительно выше среднего, что может указывать на проблемы с функцией почек. Натрий в сыворотке крови в этом кластере значительно ниже, что может быть связано с возможными нарушениями водно-солевого обмена. Люди в этом кластере в основном мужчины, курящих людей умеренно, и время наблюдения довольно низкое, что может свидетельствовать о относительно коротком периоде наблюдения для большинства из них.

Кластер 3 включает людей с возрастом, близким к среднему по набору данных, с умеренной частотой анемии и низким уровнем креатинкиназы. Частота диабета в этом кластере немного ниже средней, а фракция выброса крови близка к норме. Гипертония встречается в умеренном количестве, а уровень тромбоцитов близок к норме. Креатинин в сыворотке крови в среднем ниже, что может свидетельствовать о хорошем функционировании почек. Уровень натрия в сыворотке нормальный, а доля мужчин в этом кластере больше, чем женщин. Курящих людей умеренно, а время наблюдения выше среднего, что может указывать на более продолжительный период наблюдения за пациентами в этом кластере.

Далее будут получены метрики. См. рис. 19.

```

1 from sklearn.metrics import silhouette_score, davies_bouldin_score
2
3 # Силуэтный коэффициент
4 sil_score = silhouette_score(scaled_df[columns_to_scale], clusters)
5 print(f"Silhouette Score: {sil_score:.4f}")
6
7 # Индекс Дависа-Боулдина
8 db_index = davies_bouldin_score(scaled_df[columns_to_scale], clusters)
9 print(f"Davies-Bouldin Index: {db_index:.4f}")
✓ [321] 20ms

Silhouette Score: 0.1390
Davies-Bouldin Index: 1.8348

```

Рисунок 19 – Получение метрик

Silhouette Score: 0.1390. Это значение указывает на то, что кластеризация не является очень хорошей. Силуэтный коэффициент может варьироваться от -1 до +1, где значения близкие к +1 свидетельствуют о том, что объекты хорошо сгруппированы внутри кластеров, а значения близкие к -1 говорят о том, что объекты могут быть неправильно кластеризованы. Значение 0.1390 указывает на то, что есть неопределенность между кластерами, и кластеризация в целом не является оптимальной.

Davies-Bouldin Index: 1.8348. Этот индекс измеряет, насколько хорошо разделены кластеры. Меньшие значения говорят о лучшем разделении. Значение 1.8348 относительно высокое, что указывает на то, что кластеры не слишком хорошо разделены и существует значительная перегрузка между ними.

Кластеризация иерархическим агломеративным методом. См. рис. 20, 21, 22.

```

1  from scipy.cluster.hierarchy import dendrogram, linkage, fcluster
2  from sklearn.metrics import silhouette_score, davies_bouldin_score
3
4  Z = linkage(scaled_df[columns_to_scale], method='ward')
5
6  plt.figure(figsize=(10, 7))
7  dendrogram(Z)
8  plt.title('Дендрограмма')
9  plt.xlabel('Объекты')
10 plt.ylabel('Расстояние')
11 plt.show()
12
13 optimal_k = 4
14
15 clusters = fcluster(Z, criterion='maxclust', t=optimal_k)
16
17 cluster_means = pd.DataFrame(columns=scaled_df[columns_to_scale].columns)
18
19 for cluster_num in range(1, optimal_k + 1):
20     cluster_data = scaled_df[clusters == cluster_num]
21     cluster_means.loc[cluster_num] = cluster_data.mean()
22
23
24
25 sil_score = silhouette_score(scaled_df[columns_to_scale], clusters)
26 print(f"Silhouette Score: {sil_score:.4f}")
27
28 db_index = davies_bouldin_score(scaled_df[columns_to_scale], clusters)
29 print(f"Davies-Bouldin Index: {db_index:.4f}")

```

Рисунок 20 – Реализация метода

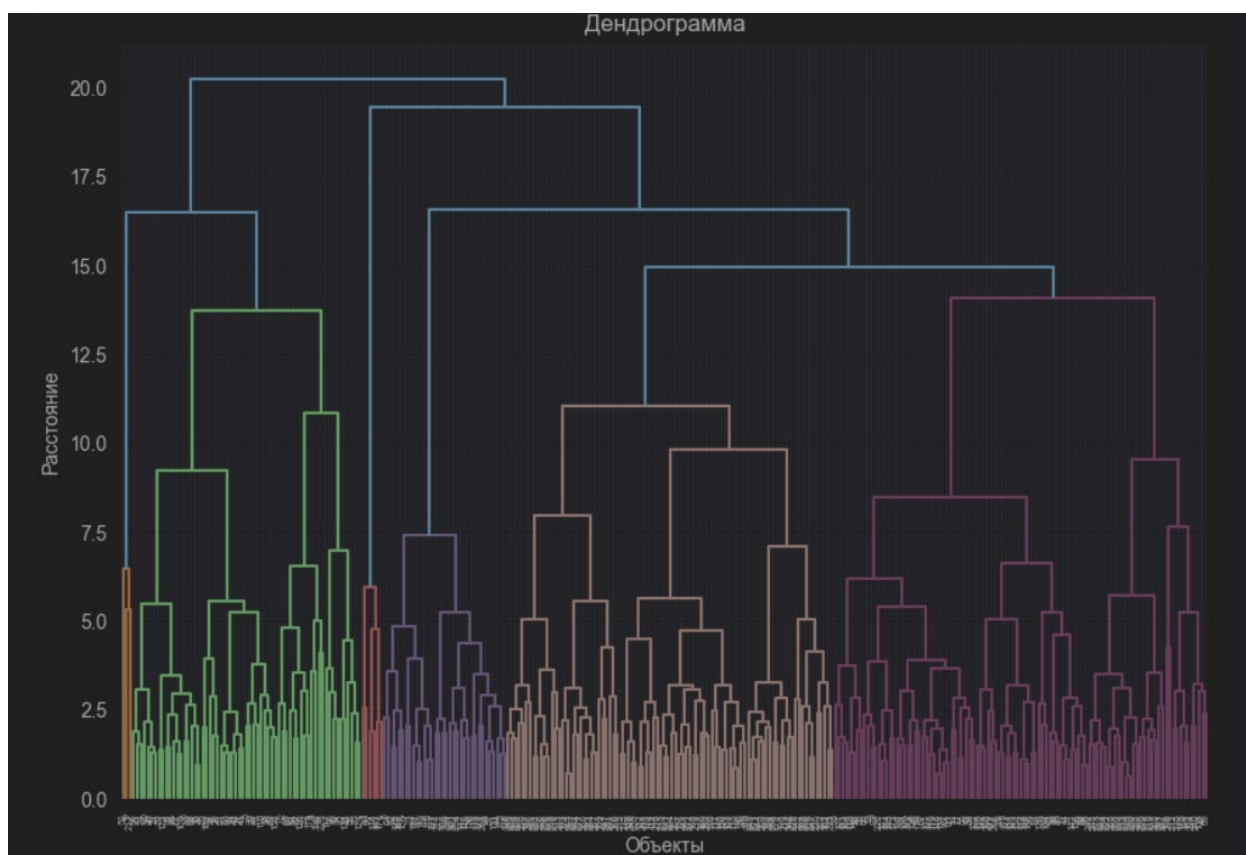


Рисунок 21 – Дендрограмма

Silhouette Score: 0.0984  
 Davies-Bouldin Index: 1.9578  
 Средние значения по каждому кластеру:

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine
1	0.773165	0.545455	-0.197701	0.424242	-0.185329	0.469697	-0.249754	0.659796
2	-0.070223	0.000000	5.448908	0.166667	-0.359735	0.166667	0.167638	-0.352331
3	0.026372	0.588235	-0.262871	0.352941	1.760362	0.441176	-0.090289	-0.334105
4	-0.266861	0.378238	-0.055480	0.435233	-0.235555	0.300518	0.096102	-0.155819

Рисунок 22- Часть таблицы и метрики

Дендрограмма показывает 6 групп, в отличие от метода локтя. Однако это можно исключить благодаря тому, что 2 кластера достаточно малы и их можно объединить с более крупными. Были получены низкие значения метрик качества кластеризации. Silhouette Score (0.0984) близок к 0, что указывает на плохое разделение объектов между кластерами, а кластеризация может быть неоптимальной. Davies-Bouldin Index (1.9578) также высок, что свидетельствует о том, что кластеры имеют большое внутреннее сходство и плохо отделены друг от друга. Эти результаты могут означать, что выбранное количество кластеров (4) не является оптимальным для данных.

Кластер 1: Этот кластер включает пациентов с относительно высоким

возрастом и умеренными показателями по анамнестическим признакам, таким как анемия и диабет. Среднее значение по `creatinine_phosphokinase` отрицательное, что может свидетельствовать о более низком уровне активности этого фермента. Пациенты из этого кластера также имеют умеренные значения по `ejection_fraction` и `high_blood_pressure`, что говорит о наличии некоторых проблем с сердечно-сосудистой системой, но без крайних отклонений. Важно отметить, что здесь больше людей с более высоким уровнем натрия в крови, но с невысоким уровнем курения и временем болезни.

Кластер 2: Пациенты из этого кластера, похоже, имеют более низкий возраст и гораздо более высокие значения по `creatinine_phosphokinase`, что может указывать на наличие серьезных проблем с сердечно-сосудистой системой или другими заболеваниями. Они также имеют относительно низкие значения по `serum_sodium` и положительные значения по признаку пола (`sex = 1`), что может свидетельствовать о том, что в этом кластере преобладают мужчины. Этот кластер также отличается по более низким значениям по `smoking` и `time`, что может говорить о более ранних стадиях заболевания.

Кластер 3: В этом кластере пациенты имеют хороший уровень `ejection_fraction`, что говорит о сохранности сердечной функции. Средние значения по признакам возраста и уровня натрия в крови показывают, что в этом кластере находятся в основном пожилые люди с хорошим уровнем контроля по сердечно-сосудистым заболеваниям. Пациенты с более высокими значениями по `high_blood_pressure` также могут быть в этом кластере, что указывает на относительно стабильное состояние с учетом сердечной активности.

Кластер 4: Этот кластер включает пациентов с более низким возрастом и умеренными показателями для большинства признаков. Здесь также наблюдаются умеренные значения для `platelets` и `serum_creatinine`, что говорит о том, что эти пациенты не имеют значительных отклонений в анализах. Они также имеют относительно нормальные уровни натрия в крови и низкие показатели по признакам `smoking` и `time`, что может свидетельствовать о более



хорошем состоянии здоровья.

#### 4. Ссылка на Google Colab:

<https://colab.research.google.com/drive/1Jx96NphL-zWkt1Jl6oD6fZZTrZ46leMt?usp=sharing>

#### 5. Вывод:

На первом этапе был выполнен предварительный анализ и стандартизация данных. Затем применены два основных метода кластеризации: алгоритм k-means и иерархическая агломеративная кластеризация. Оба метода показали, что для этого набора данных оптимальное количество кластеров — 3. С использованием метода локтя и графиков было подтверждено, что именно это количество кластеров лучше всего отражает структуру данных, в то время как использование двух или четырёх кластеров давало менее точные результаты.

В ходе работы была проведена кластеризация данных о пациентах с сердечно-сосудистыми заболеваниями для изучения различных методов обработки и анализа данных. Для этого использовались данные о возрасте, уровне креатинкиназы, фракции выброса крови и других медицинских показателях пациентов.

Для оценки качества кластеризации были использованы метрики, такие как Silhouette Score и Davies-Bouldin Index. Оба показателя показали, что разделение на кластеры не является идеальным, однако всё же даёт достаточно полезной информации о группах пациентов с различными признаками. Silhouette Score оказался низким, что говорит о неопределённости между кластерами, а Davies-Bouldin Index указывает на плохое разделение между ними. Это свидетельствует о том, что модель может быть не совсем точной, но всё же позволяет выделить некоторые ключевые группы.

В конечном итоге, полученные результаты подтверждают, что кластеризация может быть полезным инструментом для анализа таких медицинских данных, хотя для улучшения результатов стоит подумать о более сложных методах или дополнительной обработке данных.