Улядуров Андрей, 23 вариант

АМ-21-06

«Кластеризация иерархические методы и k-means»

1. Построить гистограмму распределения признаков

Импортируем и считаем параметры, выведем гистограмму распределения признаков.

In [2]:
```
import pandas as pd
from matplotlib import pyplot as plt
import numpy as np
import seaborn as sb
from sklearn import tree
from sklearn import metrics
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier, KernelDensity
from sklearn.preprocessing import StandardScaler
import warnings
warnings.filterwarnings('ignore', category = FutureWarning)
```

In [6]:
```
#импортируем наши данные из 17 листа
df = pd.read_csv(r"C:\Users\Андрей\Desktop\5 сем\Artificial_intelligence\Clustering\var23.csv", sep=';', header = 0, names = ['Pe
df.head()
```

Out[6]:

	Permeability	Total thickness	Oil-saturated thickness	Oil saturation
0	48.33	36.01	15.08	0.41
1	31.67	31.81	14.55	0.39
2	38.33	37.41	12.71	0.43
3	15.00	27.60	12.98	0.39
4	28.33	36.01	14.29	0.43

Рисунок 1. Считываем признаки, выводим "голову" датафрейма

In [7]:
```
#гистограмма распределения признаков до стандартизации
plt.hist(df)
plt.legend(['Permeability', 'Total thickness', 'Oil-saturated thickness', 'Oil saturation'])
```

Out[7]:
```
<matplotlib.legend.Legend at 0x246fb5a2800>
```

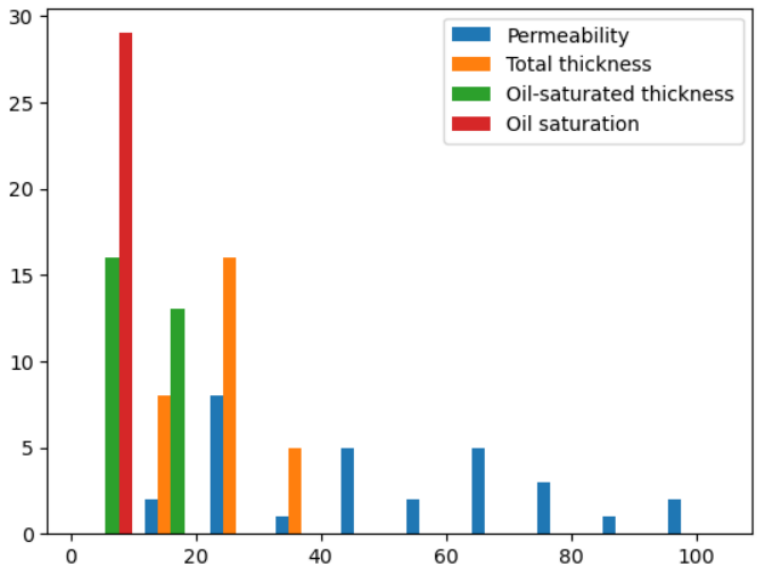


Рисунок 2. Гистограмма распределения признаков

2. Провести нормализацию или стандартизацию значений параметров

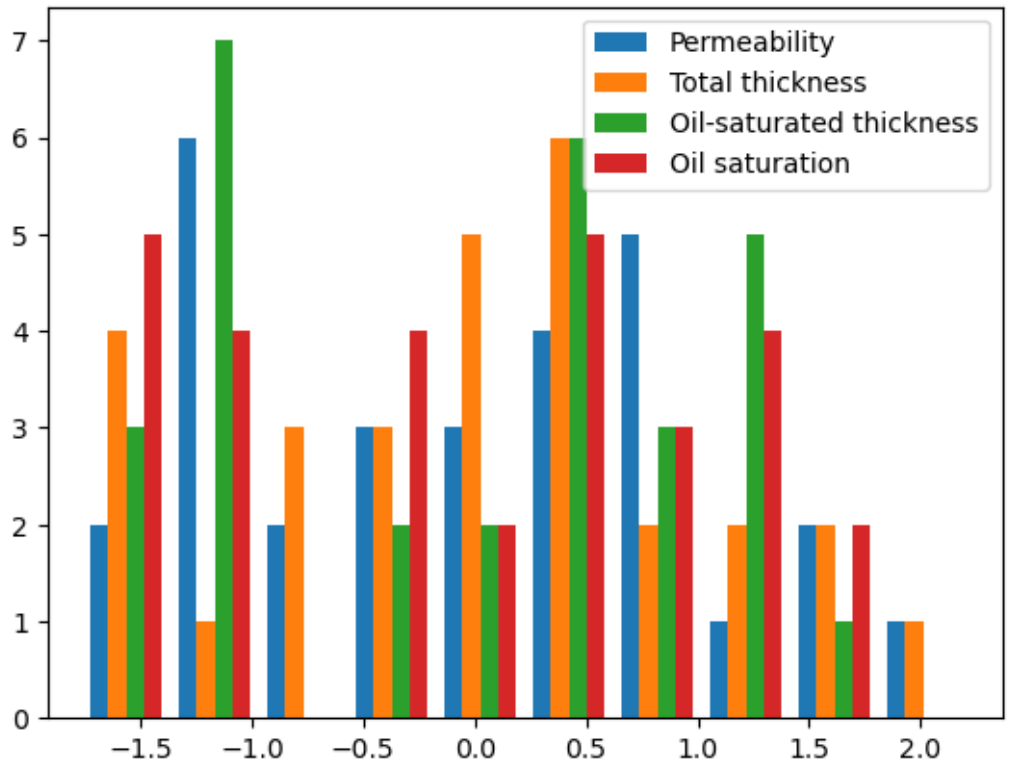


Рисунок 3. Гистограмма отнорированных и стандартизовнных признаков

3. Построить 3d график распределения объектов с нормированными/стандартизированными характеристиками (одну из характеристик придется исключить, чтобы получить 3 координаты, но из кластеризации ее не исключаем). Чтобы понять, какую характеристику лучше НЕ отображать на 3d графике, нужно вычислить корреляционную матрицу. На графике можно исключить из координат тот столбец, который имеет максимальные по модулю значения коэффициента корреляции

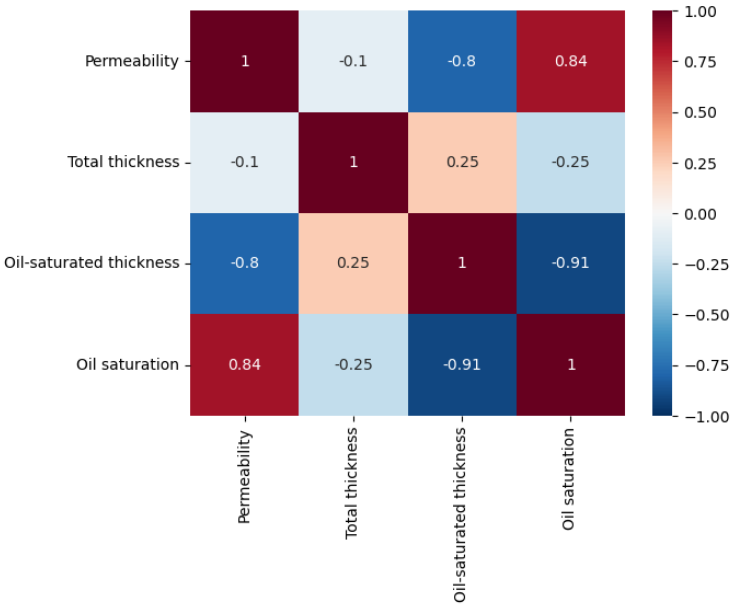


Рисунок 4. Ковариационная матрица признаков

Исключим признак «Oil saturation»

Построим 3-d графике, эти три признака, которые остались.

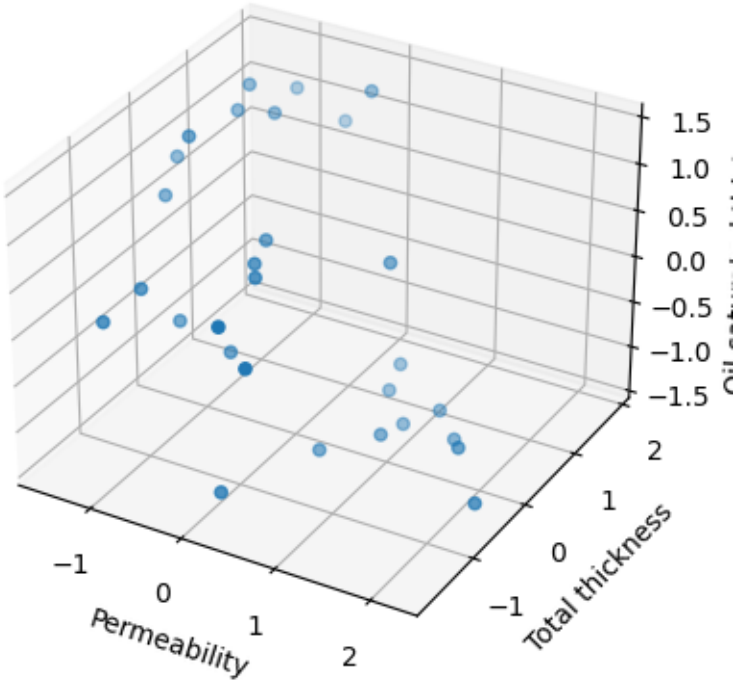
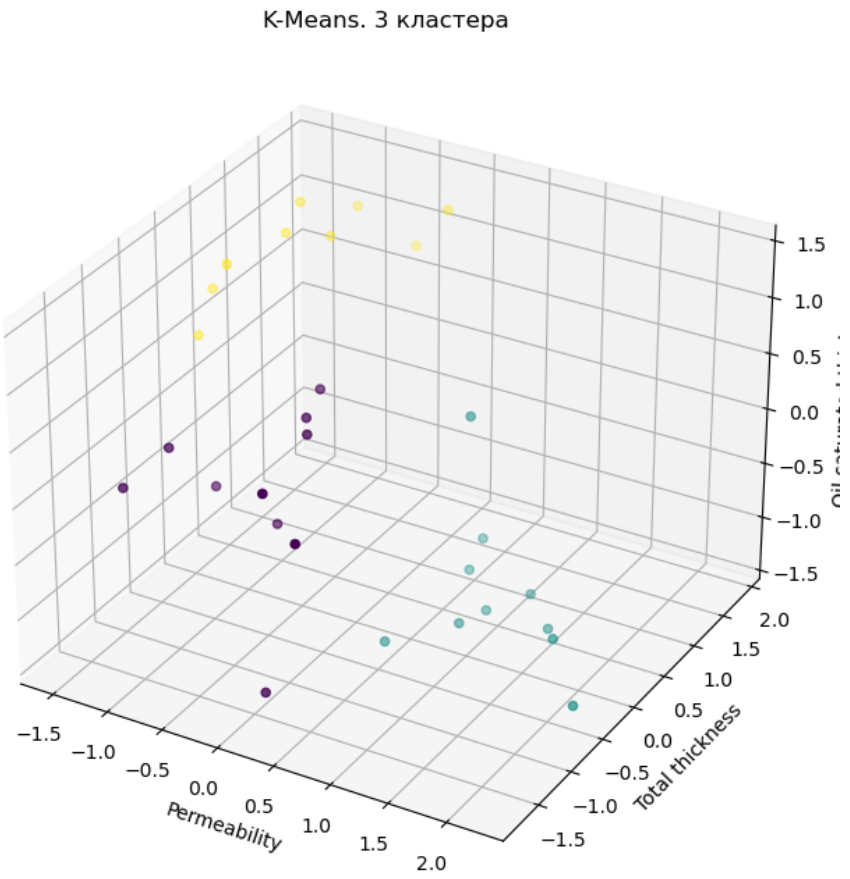


Рисунок 5. 3д График признаков "Permeability", "Total thickness", "Oil saturation"

4. Получить разбиение объектов на кластеры 3 разными иерархическими методами и методом k-means. Попробуйте в одном из методов использовать расстояние Чебышева. 5. Построить дендрограммы для иерархических методов. 6. Сравнить результаты полученные при разбиении для различных методов, сделать вывод об устойчивости разбиения. Результаты кластеризации показать в таблице. Пример таблицы

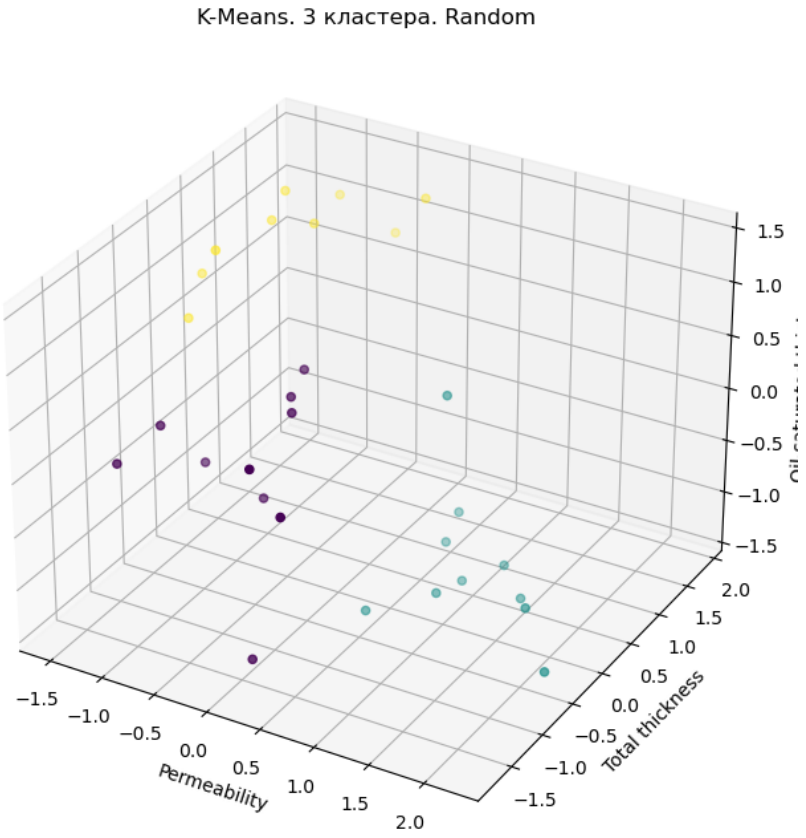
1) K-means:

```
clstr1 = KMeans(n_clusters = 3, algorithm = 'lloyd')
```



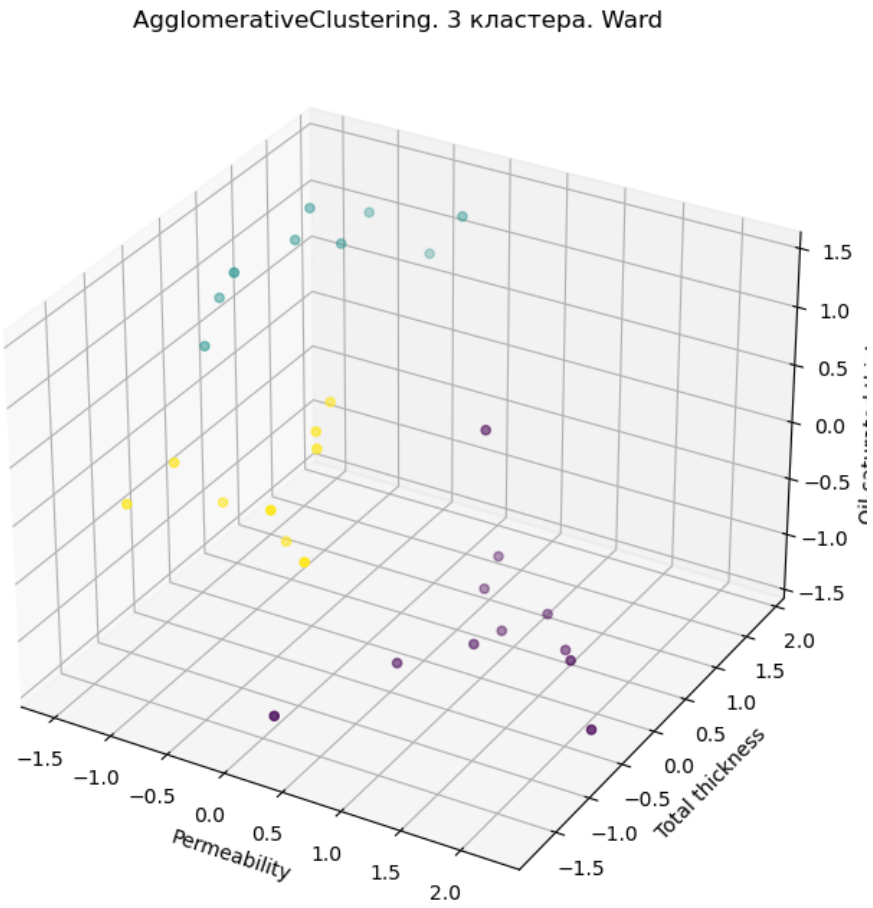
K-means с методом ‘random’:

```
KMeans(n_clusters = 3, init='random', algorithm = 'lloyd')
```



2) Agglomerative-clustering:

```
AgglomerativeClustering(n_clusters=3, linkage='ward')
```



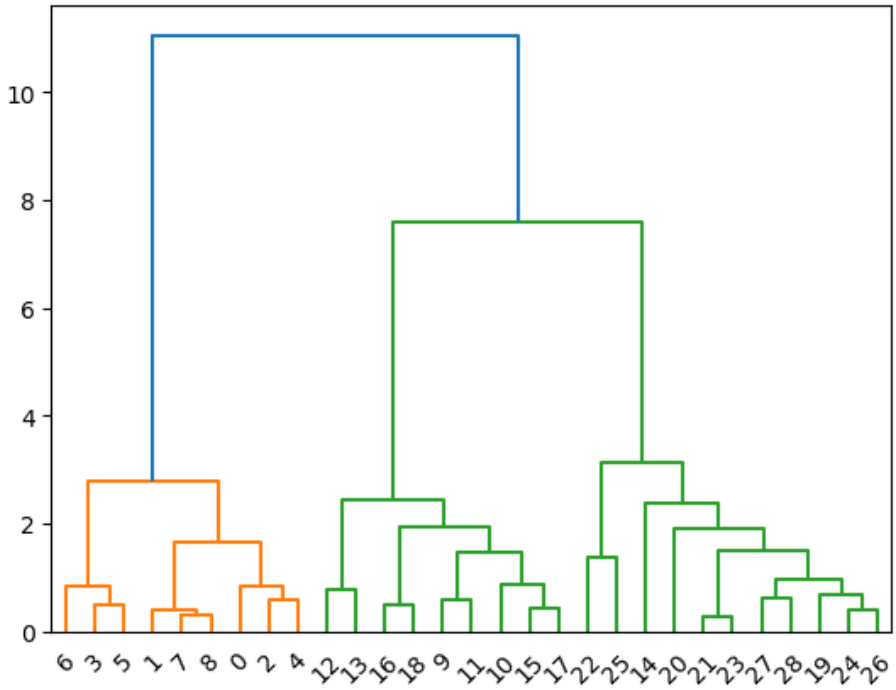


Рисунок 6. Дендрограмма AgglomerativeClustering, ward метод

3) AgglomerativeClustering с average методом и manhattan метрикой:

```
AgglomerativeClustering(n_clusters=3, linkage='average', metric ='manhattan')
```

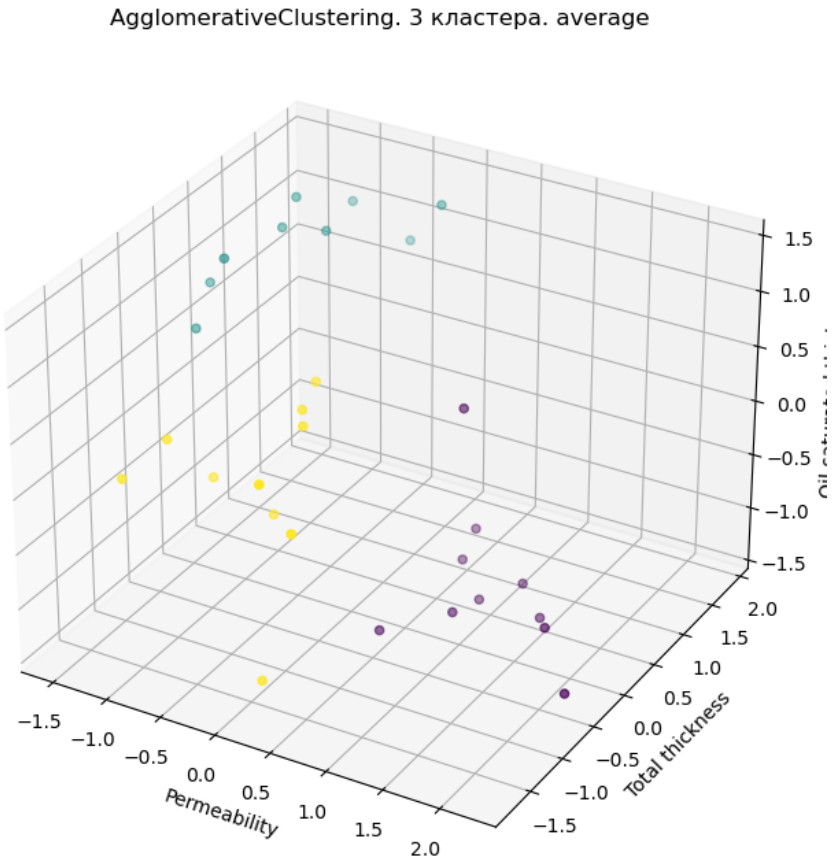


Рисунок 7. AgglomerativeClustering, average метод, manhattan метрика

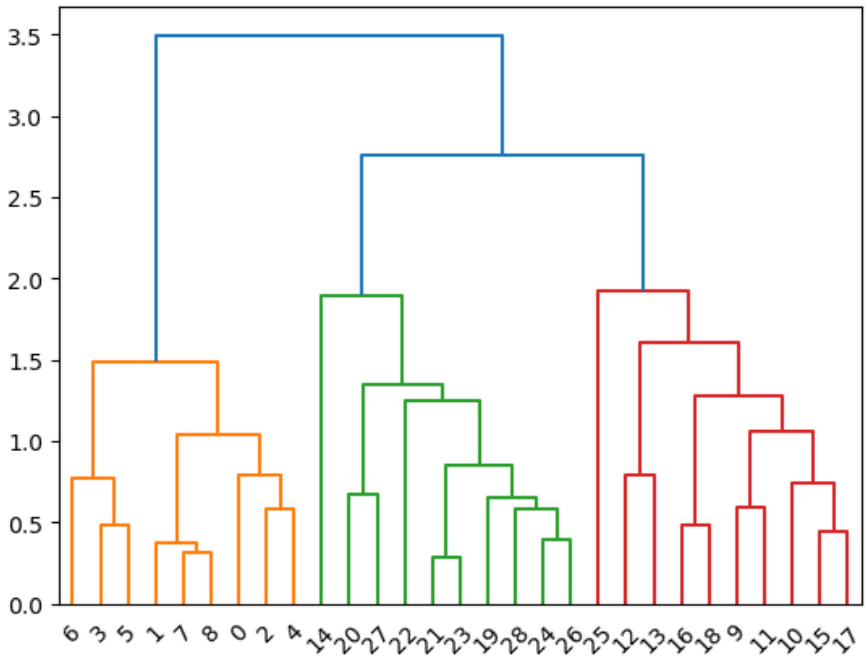


Рисунок 8. AgglomerativeClustering, average метод, manhattan метрика, дендрограмма

4) AgglomerativeClustering с average методом и метрикой Чебышева:

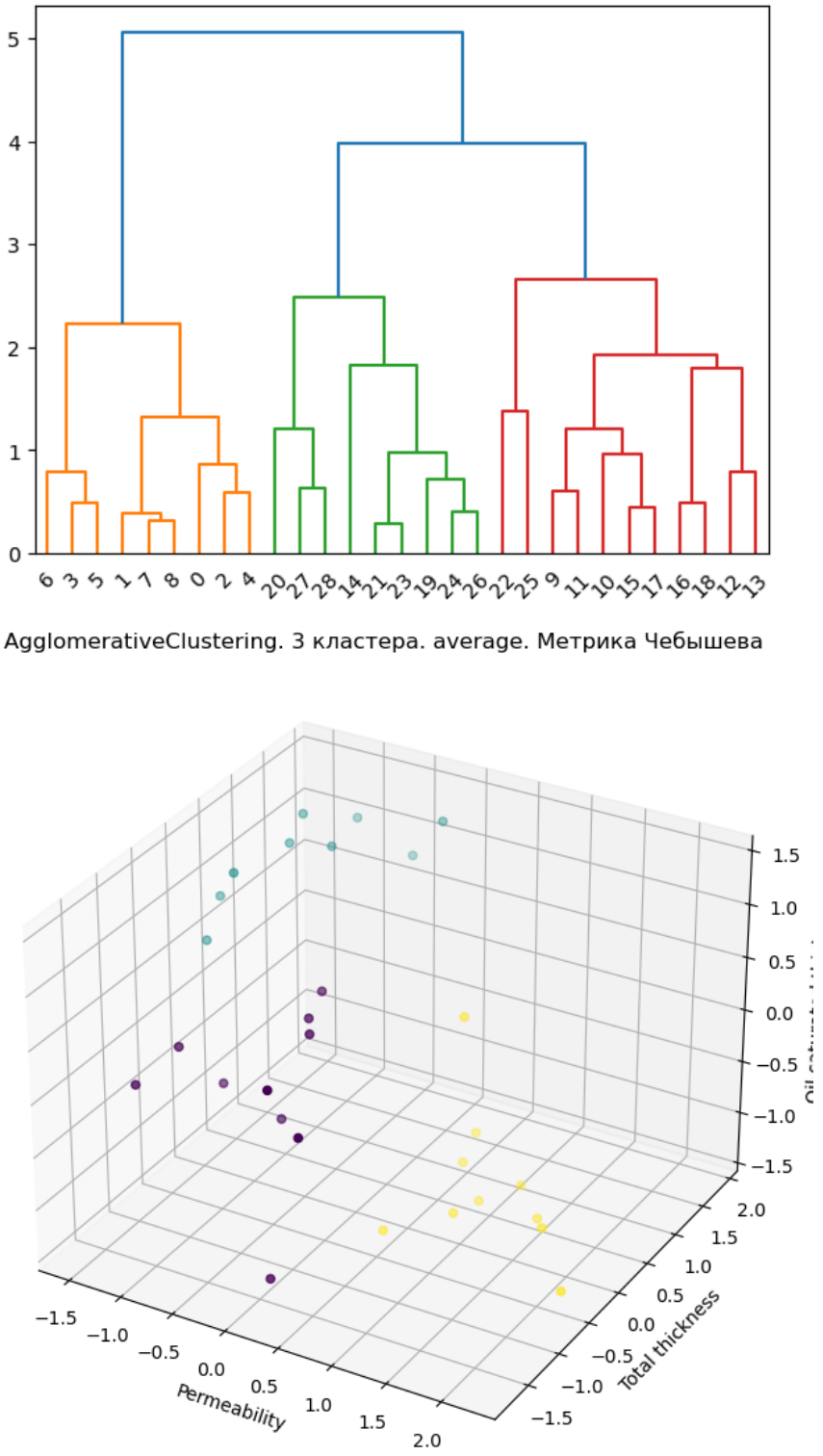


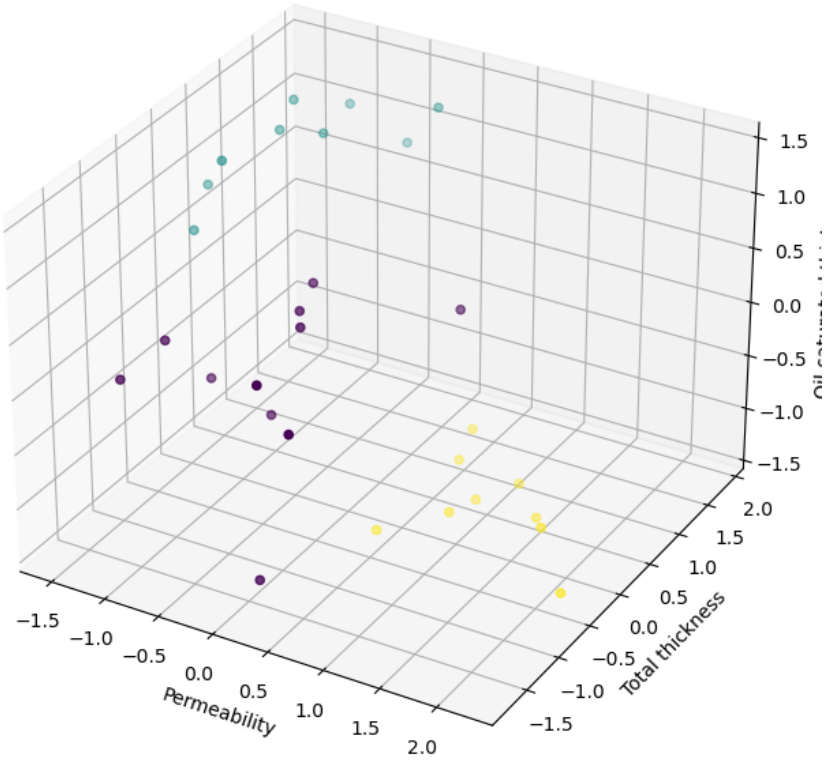
Рисунок 9. AgglomerativeClustering, average метод, метрика Чебышева, дендрограмма

Итоговая таблица способов кластеризации:

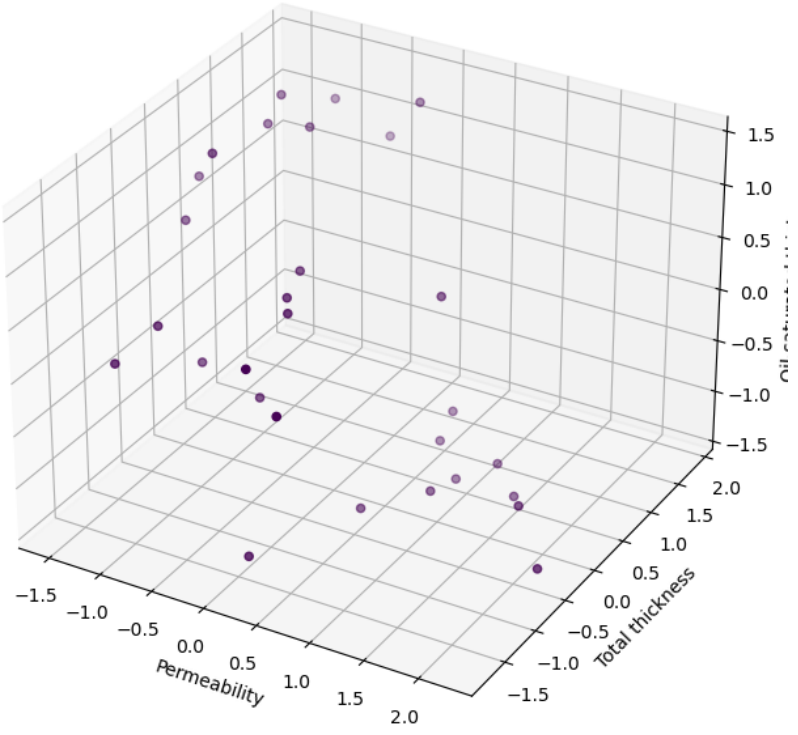
	Kmeans, инициализация k-means++	Kmeans, инициализация random	Метод ward, расстояние Евклидово	Метод Average, расстояние Manhattan	Метод Average, расстояние Чебышева
0	0	2	1	1	1
1	0	2	1	1	1
2	0	2	1	1	1
3	0	2	1	1	1
4	0	2	1	1	1
5	0	2	1	1	1
6	0	2	1	1	1
7	0	2	1	1	1
8	0	2	1	1	1
9	1	0	2	2	0
10	1	0	2	2	0
11	1	0	2	2	0
12	1	0	2	2	0
13	1	0	2	2	0
14	2	1	0	0	2
15	1	0	2	2	0
16	1	0	2	2	0
17	1	0	2	2	0
18	1	0	2	2	0
19	2	1	0	0	2
20	2	1	0	0	2
21	2	1	0	0	2
22	2	1	0	0	2
23	2	1	0	0	2
24	2	1	0	0	2
25	1	0	0	2	0
26	2	1	0	0	2
27	2	1	0	0	2
28	2	1	0	0	2

5) И, наконец, вариации DBSCAN алгоритма:

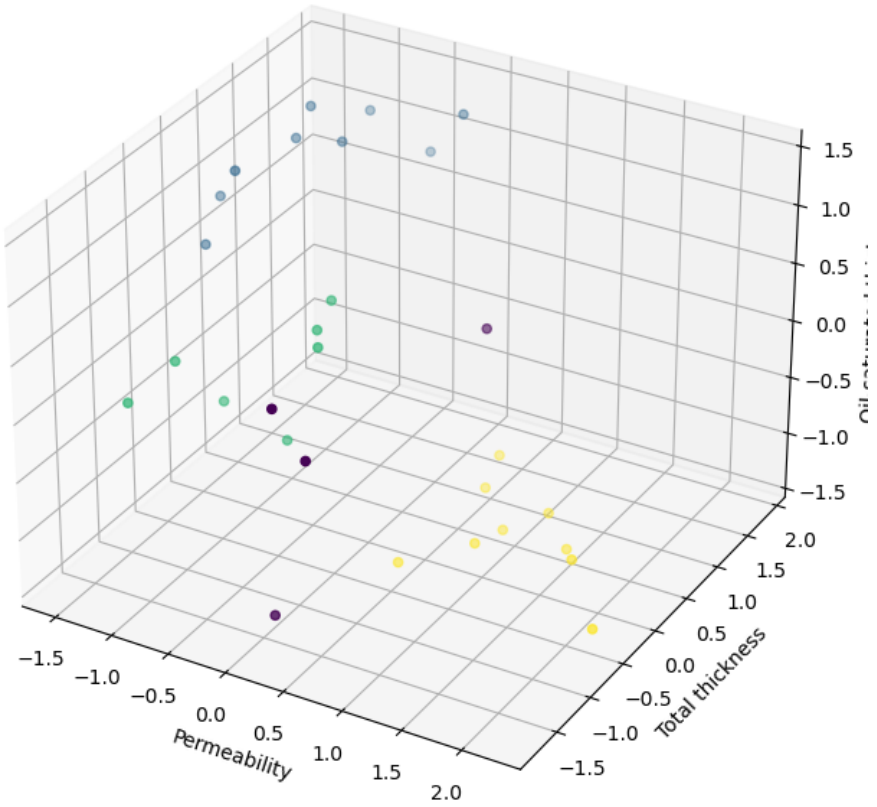
```
DBSCAN(eps=1, min_samples=5) :
```



`DBSCAN(eps=0.5, min_samples=5)`:



`DBSCAN(eps=1.1, min_samples=3)`:



DBSCAN:

	eps=1, min_samples=5	eps=0.5, min_samples=5	eps=1.1, min_samples=3
0	0	-1	0
1	0	-1	0
2	0	-1	0
3	0	-1	0
4	0	-1	0
5	0	-1	0
6	0	-1	0
7	0	-1	0
8	0	-1	0
9	-1	-1	1
10	-1	-1	1
11	-1	-1	1
12	-1	-1	-1
13	-1	-1	-1
14	-1	-1	-1
15	-1	-1	1
16	-1	-1	1
17	-1	-1	1
18	-1	-1	1

19	1	-1	2
20	1	-1	2
21	1	-1	2
22	1	-1	2
23	1	-1	2
24	1	-1	2
25	-1	-1	-1
26	1	-1	2
27	1	-1	2
28	1	-1	2

Выводы:

Как видно из таблиц, для метрик manhattan и метрики Чебышева при методе average, деление на кластеры для наших данных происходит одинаковое, и это деление похоже на метод ward при Евклидовом расстоянии. Это может говорить об устойчивости наших данных. K-means делит похожим на предыдущие методы способом, что говорит о «разрозненности» наших данных. Можно заметить, что элемент, не совпадающий при делении Kmeans и иных делениях – находится примерно на одинаковой удаленности от двух близких кластеров, и попадает в разные в силу специфики каждого алгоритма. Также рассмотрено деление DBSCAN, которое не зависит от изначального количества заданных нами кластеров, а само определяет это число. Видно, что с параметрами eps=0.5, min_samples=5 он заталкивает все точки в один класс, скорее всего это связанно с тем, что eps слишком мал, как максимальное расстояние между двумя точками, чтобы одна из них считалась соседней с другой. Также видно, что при указании больших значений min_samples, DBSCAN найдет более плотные кластеры, то есть большее количество кластеров (при сравнении 3 столбца и 1ого). При указании низких значений min_samples – более разреженные кластеры, то есть меньшее количество кластеров.