Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Пермский государственный национальный исследовательский
университет»

Цифровая кафедра

Отчёт по работе

«Анализ факторов, влияющих на расходы покупателей в магазине»

Выполнили:

Студенты 4-го курсов

физико-математического института факультета

Т. М. Овчинников

М. Л. Мазязин

Проверил:

к.ф.-м.н. А. В. Ратт

г. Пермь 2024

Содержание

Введение

Расходы покупателей являются важным аспектом функционирования рынка и экономики в целом. Они влияют на потребительский спрос, распределение доходов и уровень благосостояния населения. Изучение расходов покупателей позволяет понять закономерности поведения потребителей, определить их предпочтения и потребности, а также разработать стратегии для удовлетворения этих потребностей и стимулирования спроса на определённые товары и услуги.

В современном мире, где рынок становится всё более конкурентным, компаниям необходимо учитывать расходы своих потенциальных клиентов, чтобы успешно конкурировать и привлекать внимание потребителей. Изучение расходов покупателей также помогает выявить возможные тенденции и изменения в потребностях и предпочтениях аудитории, что позволяет компаниям адаптировать свои продукты и услуги в соответствии с этими изменениями.

Цель работы: проанализировать данные о клиентах магазина, выполнить разбиение данных на кластеры и оценить качество кластеризации.

Техническое задание: требуется проанализировать данные о клиентах магазина (файл Customers.csv), выявить зависимости между факторными переменными, разбить данные на кластеры. Дать интерпретацию полученным результатам. Сделать выводы.

1. Подготовка данных к анализу

Была выполнена загрузка данных в датафрейм:

```
data = pd.read_csv("Customers.csv")
data
```

	CustomerID	Gender	Age	Annual Income ($)	Spending Score (1-100)	Profession	Work Experience	Family Size
0	1	Male	19	15000	39	Healthcare	1	4
1	2	Male	21	35000	81	Engineer	3	3
2	3	Female	20	86000	6	Engineer	1	1
3	4	Female	23	59000	77	Lawyer	0	2
4	5	Female	31	38000	40	Entertainment	2	6
...	...	...	...	...	...	...	...	...
1995	1996	Female	71	184387	40	Artist	8	7
1996	1997	Female	91	73158	32	Doctor	7	7
1997	1998	Male	87	90961	14	Healthcare	9	2
1998	1999	Male	77	182109	4	Executive	7	2
1999	2000	Male	90	110610	52	Entertainment	5	2

Рисунок 1. Данные датафрейма

Далее выполнена проверка, что все количественные столбцы имеют числовой тип, с последующим преобразованием типа столбца к числовому. А также проверка на пропуски, строки с пропусками были удалены из датафрейма:

```
data
```

	CustomerID	Gender	Age	Spending Score (1-100)	Profession	Work Experience	Family Size	Annual Income (K$)
0	1	1	19	39	5	1	4	15.000
1	2	1	21	81	2	3	3	35.000
2	3	0	20	6	2	1	1	86.000
3	4	0	23	77	7	0	2	59.000
4	5	0	31	40	3	2	6	38.000
...	...	...	...	...	...	...	...	...
1995	1996	0	71	40	0	8	7	184.387
1996	1997	0	91	32	1	7	7	73.158
1997	1998	1	87	14	5	9	2	90.961
1998	1999	1	77	4	4	7	2	182.109
1999	2000	1	90	52	3	5	2	110.610

Рисунок 2. Данные датафрейма после подготовки

2. Предварительный анализ данных

В предварительном анализе данные было произведено вычисление описательных статистик по колонкам (среднее, моду, медиану, стандартное отклонение, квартили):

```
data.describe()
```

	CustomerID	Gender	Age	Spending Score (1-100)	Profession	Work Experience	Family Size	Annual Income (K$)
count	1965.000000	1965.000000	1965.000000	1965.000000	1965.000000	1965.000000	1965.000000	1965.00000
mean	1000.309924	0.406616	48.894656	51.078880	2.830534	4.092621	3.757252	110.61601
std	578.443714	0.491327	28.414889	27.977176	2.544969	3.926459	1.968335	45.83386
min	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	0.00000
25%	498.000000	0.000000	25.000000	28.000000	0.000000	1.000000	2.000000	74.35000
50%	1000.000000	0.000000	48.000000	50.000000	3.000000	3.000000	4.000000	109.75900
75%	1502.000000	1.000000	73.000000	75.000000	5.000000	7.000000	5.000000	149.09500
max	2000.000000	1.000000	99.000000	100.000000	8.000000	17.000000	9.000000	189.97400

Рисунок 3. *Описательные статистики*

Далее проверка числовых колонок на наличие выбросов, для этого можно использовать диаграмму «ящик с усами» (boxplot), у которого был выявлен выброс на «Work experience»:

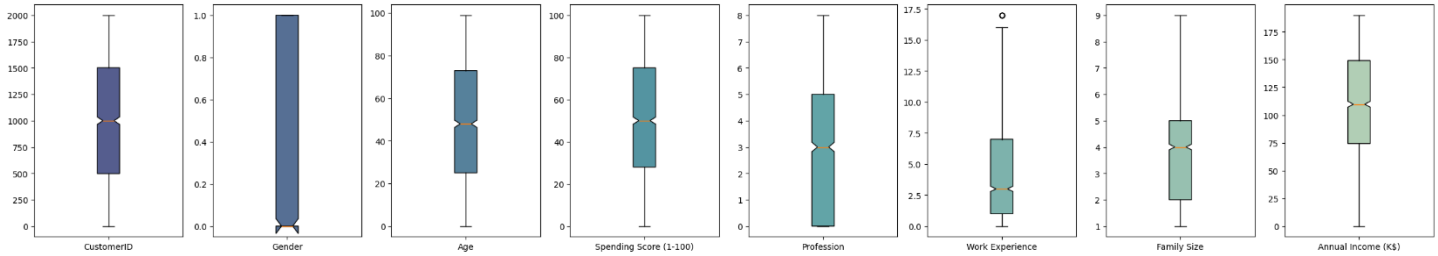

Рисунок 4. *Ящики с усами*

От выброса можно избавиться заменой на 99-ый процентиль:

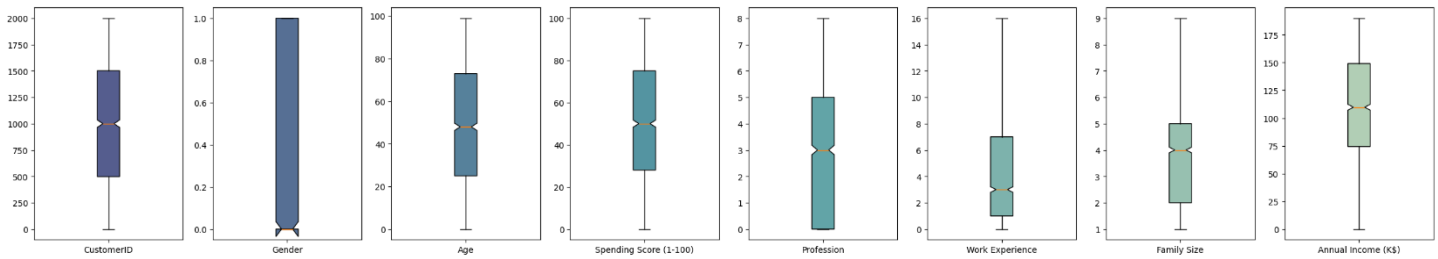

Рисунок 5*. Ящики с усами без выбросов*

После произведена проверка колонок на нормальность распределения с помощью:

- Гистограммы рассеяния

- статического теста

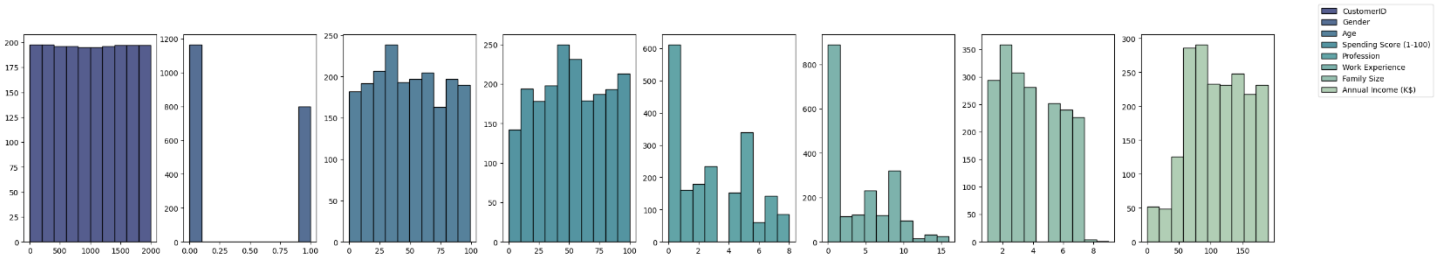

Рисунок 6. Гистограммы

Но ни на одной гистограмме не наблюдается нормального распределения, а для статистического тест была следующая нулевой гипотезой: ***«Распределение столбцов является нормальным»:***

```
for i, p in enumerate(sp.stats.normaltest(data).pvalue):
    print(f"p-критерий для столбца {data.columns[i]}: {p}\n")
```

```
p-критерий для столбца CustomerID: 0.0

p-критерий для столбца Gender: 0.0

p-критерий для столбца Age: 6.296095611304533e-266

p-критерий для столбца Spending Score (1-100): 6.549544156026916e-169

p-критерий для столбца Profession: 6.28626017578814e-153

p-критерий для столбца Work Experience: 2.6173974591519126e-38

p-критерий для столбца Family Size: 3.3789479938013944e-252

p-критерий для столбца Annual Income (K$): 6.986109218690106e-46

p-критерий для каждого из столбцов стремится к нулю
```

Рисунок 7. Статистический тест

Из теста видно, что p-критерий для каждого из столбцов стремится к нулю, а значит: ***«Нулевая гипотеза не выполняется, столбцы не имеют нормального распределения».***

Была построена корреляционная матрица с последующим отбором признаков для кластеризации – такие, которые как можно меньше зависят друг от друга.

Результатом **предварительного анализа данных** должна быть выдвинутая гипотеза о том, что данные можно разделить на некоторое количество кластеров (предположить, каково это количество), используя для группировки отобранные признаки и произведём нормализацию выбранных для кластеризации признаки:

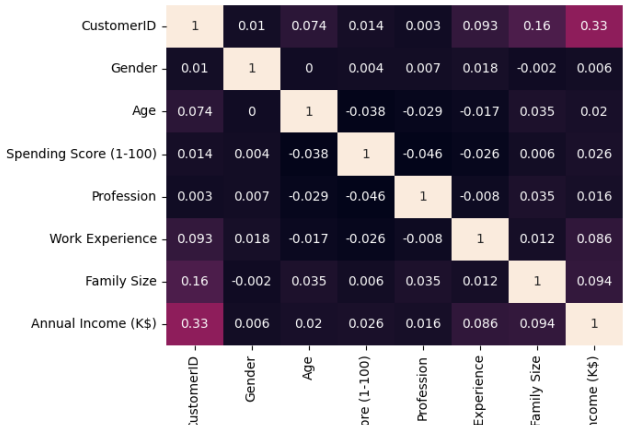

Рисунок 8. Корреляционная матрица с признаками для кластеризации

```
from sklearn import preprocessing as prcss
data_scaled = pd.DataFrame(
    prcss.MinMaxScaler().fit_transform(data[data.columns[1::]]),
    columns = data.columns[1::]
)
data_scaled
```

	Gender	Age	Spending Score (1-100)	Profession	Work Experience	Family Size	Annual Income (K$)
0	1.0	0.191919	0.39	0.625	0.0625	0.375	0.078958
1	1.0	0.212121	0.81	0.250	0.1875	0.250	0.184236
2	0.0	0.202020	0.06	0.250	0.0625	0.000	0.452694
3	0.0	0.232323	0.77	0.875	0.0000	0.125	0.310569
4	0.0	0.313131	0.40	0.375	0.1250	0.625	0.200027
...	...	...	...	...	...	...	...
1960	0.0	0.717172	0.40	0.000	0.5000	0.750	0.970591
1961	0.0	0.919192	0.32	0.125	0.4375	0.750	0.385095
1962	1.0	0.878788	0.14	0.625	0.5625	0.125	0.478808
1963	1.0	0.777778	0.04	0.500	0.4375	0.125	0.958600
1964	1.0	0.909091	0.52	0.375	0.3125	0.125	0.582238

Рисунок 9. Нормализация для кластеризации признаки

Из полученных данных можно сделать вывод, что данные можно разделить на ***2, 9 и 17 кластеров***, используя признаки: Gender, Age, Spending Score (1-100), Profession, Work Experience, Family Size, Annual Income (K$). 2 по числу полов, 9 по числу профессий и 17 по опыту работы.

3. Кластерный анализ

Осуществлена кластеризация данных методом кластеризации K-means с предположенным ранее числом кластеров (17 по опыту работы).

```
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score as s_score
```

```
kmeans = KMeans(17, init='k-means++', algorithm='elkan')
kmeans.fit(data_scaled)
```

```
kmeans_gl = KMeans(17, init='k-means++', algorithm='elkan')
kmeans_gl.fit(data_scaled[1::])
```

Рисунок 10. Метод K-means

Вычислим для него силуэт:

```
s_score(data_scaled, kmeans.predict(data_scaled))
```

```
0.14260517077572077
```

```
s_score(data_scaled[1::], kmeans_gl.predict(data_scaled[1::]))
```

```
0.141960967574707
```

Рисунок 11. Метод K-means

Найдем количество кластеров, при котором значение силуэта максимально:

```
clast = []
silhs = []
for i in np.arange(2, 18):
    kmeans = KMeans(i, init='k-means++', algorithm='lloyd')
    kmeans.fit(data_scaled)
    clast.append(i)
    silhs.append(s_score(data_scaled, kmeans.predict(data_scaled)))
    print(f'Количество кластеров: {i}:', silhs[-1])
```

```
Количество кластеров: 2: 0.33586090607117003
Количество кластеров: 3: 0.2297158191606437
Количество кластеров: 4: 0.1702940237230681
Количество кластеров: 5: 0.16331166666829278
Количество кластеров: 6: 0.1444400938498799
Количество кластеров: 7: 0.13828835463634392
Количество кластеров: 8: 0.14603186765962384
Количество кластеров: 9: 0.14029696875596603
Количество кластеров: 10: 0.13819480467051853
Количество кластеров: 11: 0.14030701479962243
Количество кластеров: 12: 0.13888283695214856
Количество кластеров: 13: 0.14152767703899094
Количество кластеров: 14: 0.13732468613226875
Количество кластеров: 15: 0.13761966795743868
Количество кластеров: 16: 0.14280869533230317
Количество кластеров: 17: 0.13344566457030277
```

Рисунок 12. Значение силуэта от количества кластеров

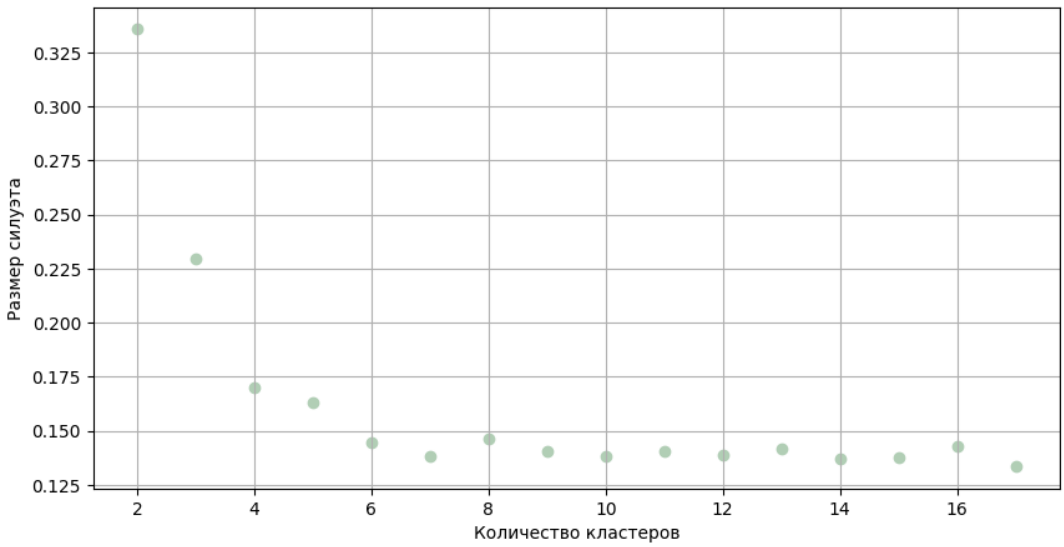

Рисунок 13. Зависимость силуэта от количества кластеров

Рассмотрим эти кластеры:

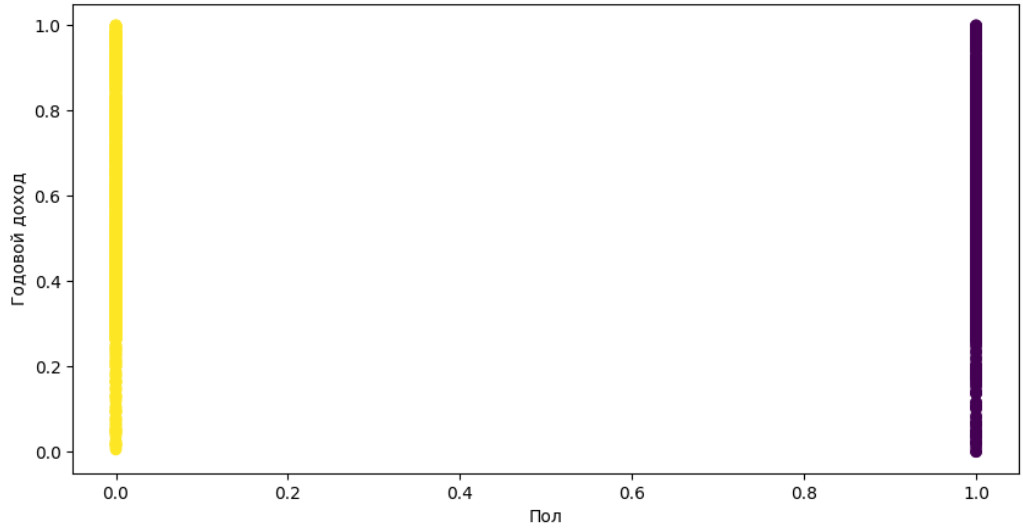

Рисунок 14. Количество кластеров 2

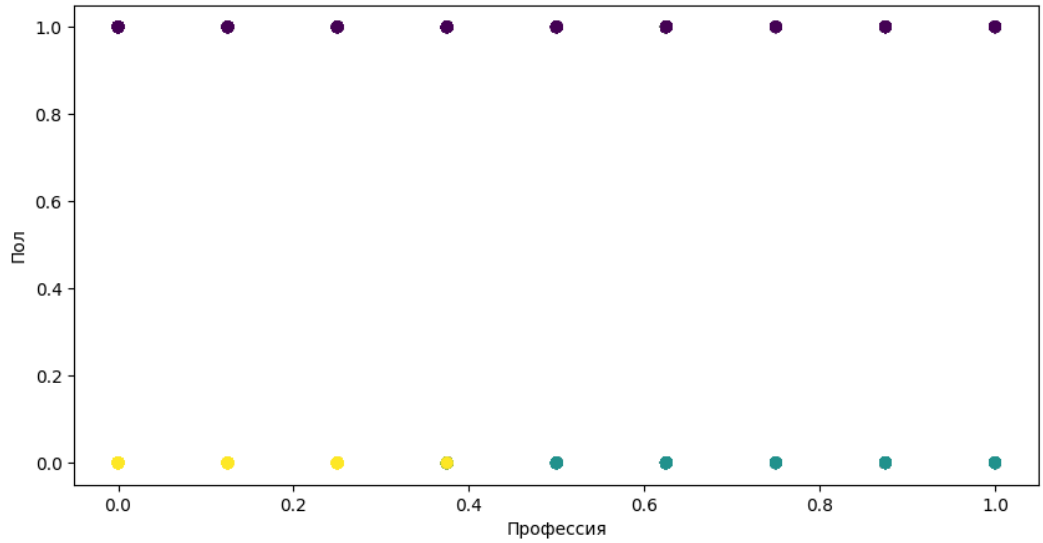

Рисунок 15. Количество кластеров 3

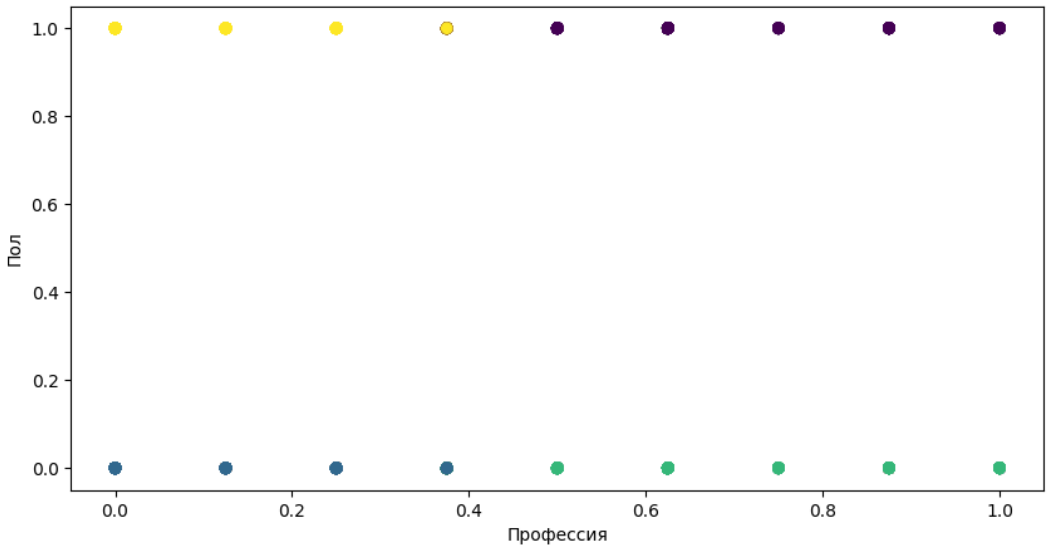

Рисунок 16. Количество кластеров 4

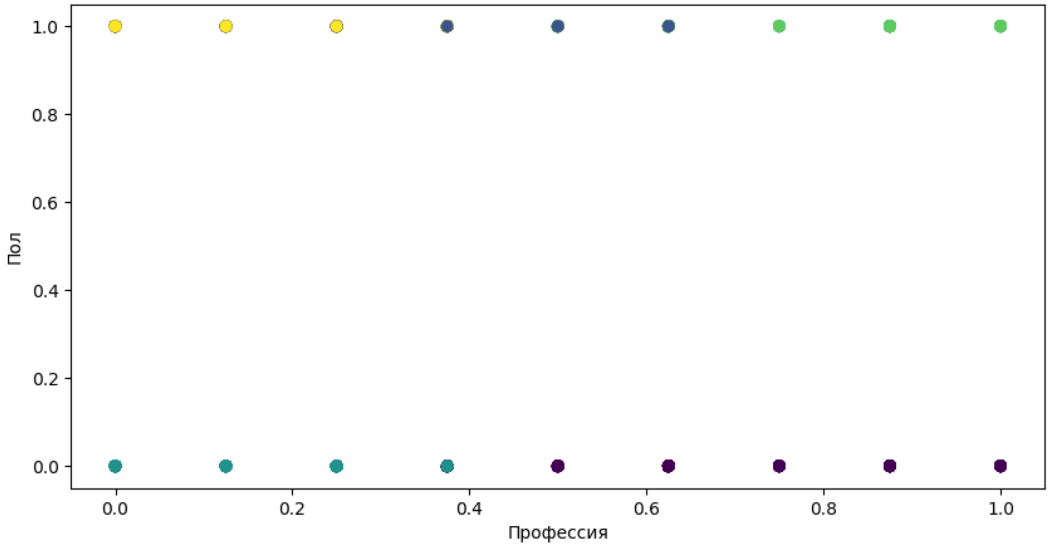

Рисунок 17. Количество кластеров 5

Также осуществлена кластеризация данных методом кластеризации SpectralClustering:

```
from sklearn.cluster import SpectralClustering
```

```
clast = []
silhs = []
for i in np.arange(2, 18):
    spcl = SpectralClustering(i)
    spcl.fit(data_scaled)
    clast.append(i)
    silhs.append(s_score(data_scaled, spcl.fit_predict(data_scaled)))
    print(f'Количество кластеров: {i}:', silhs[-1])
```

Рисунок 18. Метод кластеризации SpectralClustering и значение силуэта от количества кластеров

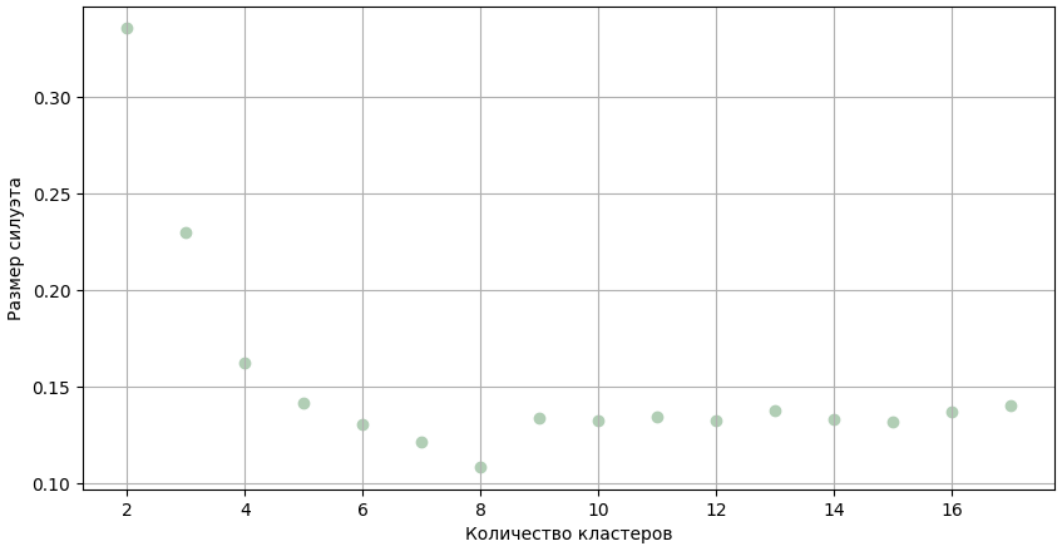

Рисунок 19. Зависимость силуэте от количества кластеров

Рассмотрим эти кластеры:

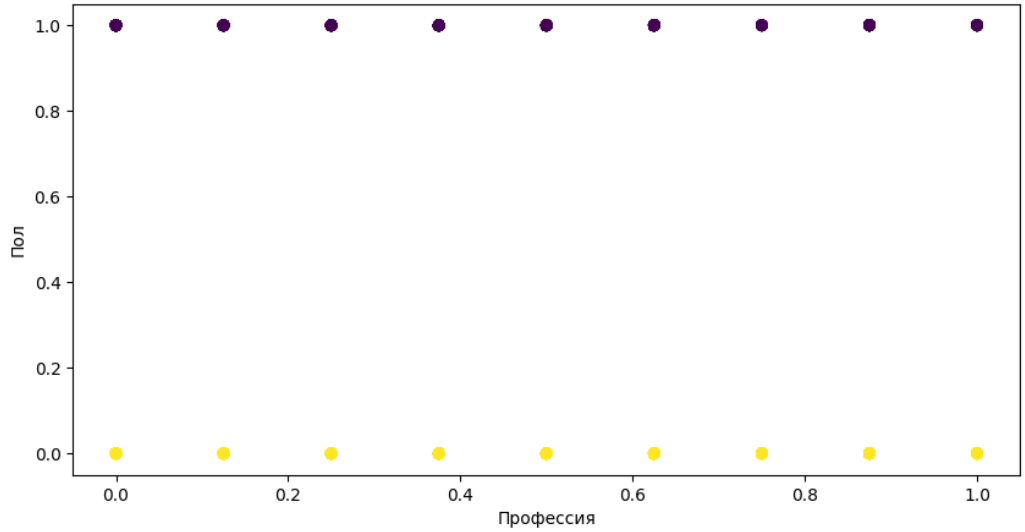

Рисунок 20. Количество кластеров 2

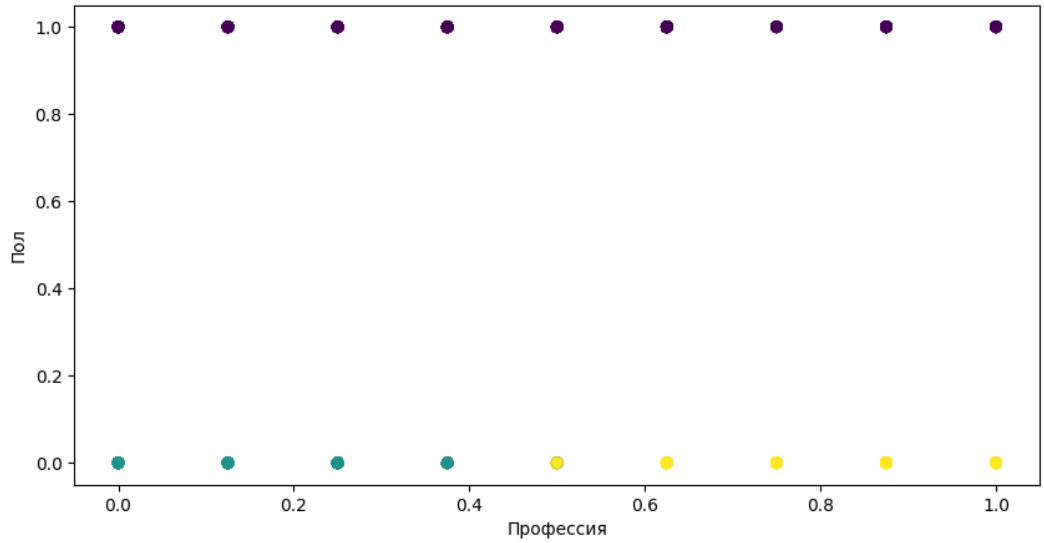

Рисунок 21. Количество кластеров 3

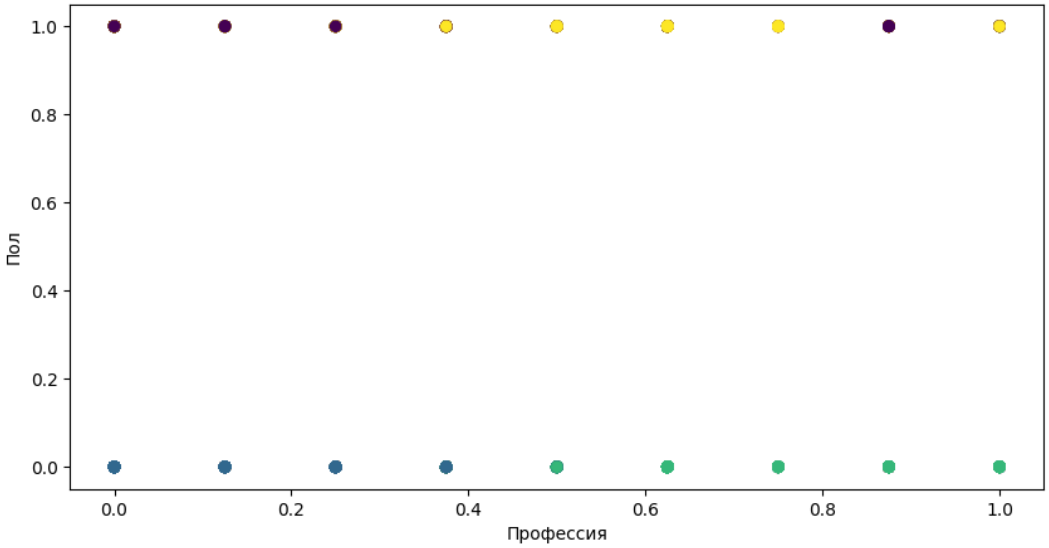

Рисунок 22. Количество кластеров 4

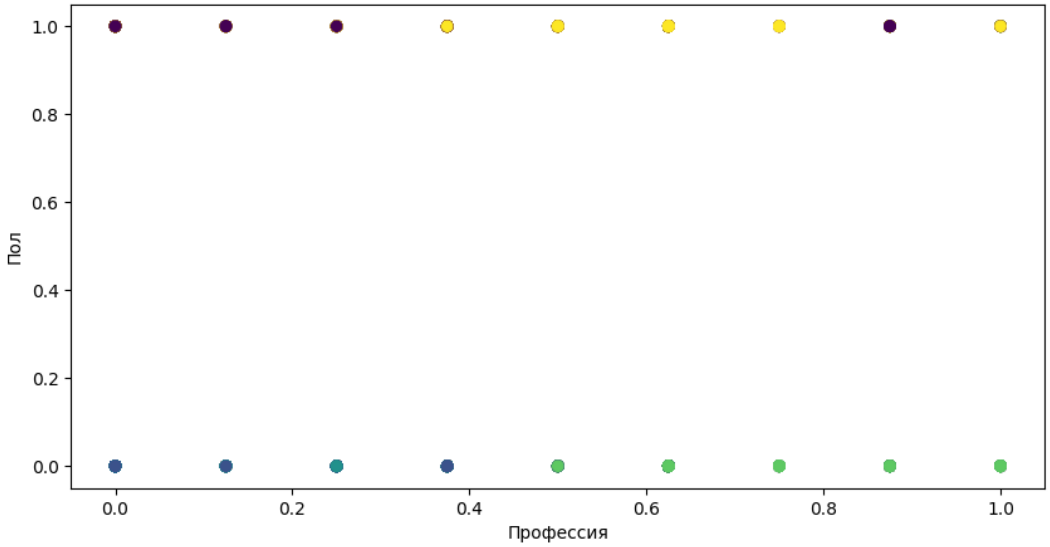

Рисунок 23. Количество кластеров 5

Осуществлена кластеризация данных методом кластеризации BisectingKMeans:

```
from sklearn.cluster import BisectingKMeans
```

```
clast = []
silhs = []
for i in np.arange(2, 18):
    spcl = BisectingKMeans(i)
    spcl.fit(data_scaled)
    clast.append(i)
    silhs.append(s_score(data_scaled, spcl.fit_predict(data_scaled)))
    print(f'Количество кластеров: {i}:', silhs[-1])
```

Рисунок 24. Метод кластеризации BisectingKMeans и значение силуэта от количества кластеров

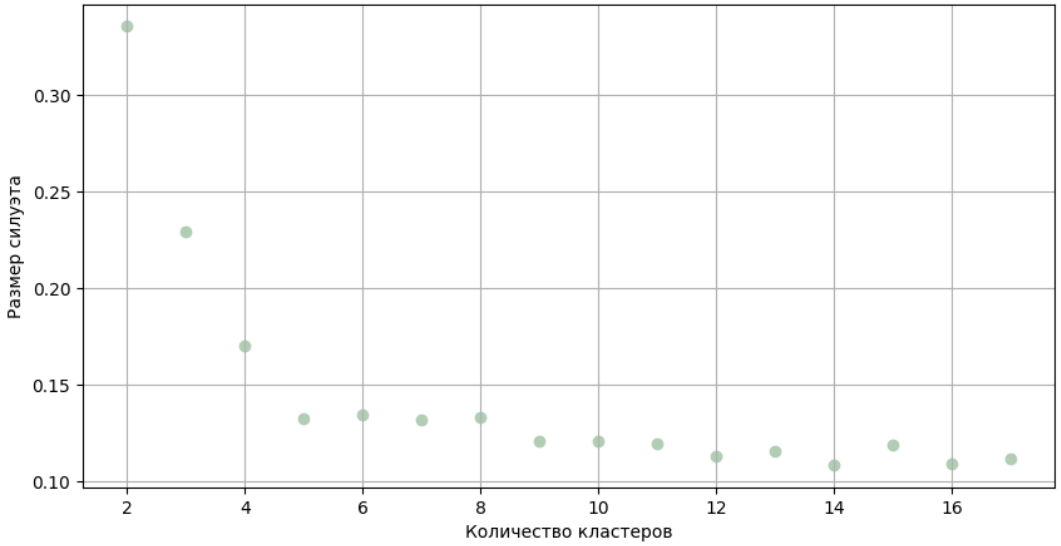

Рисунок 25. Зависимость силуэта от количества кластеров

Рассмотрим эти кластеры:

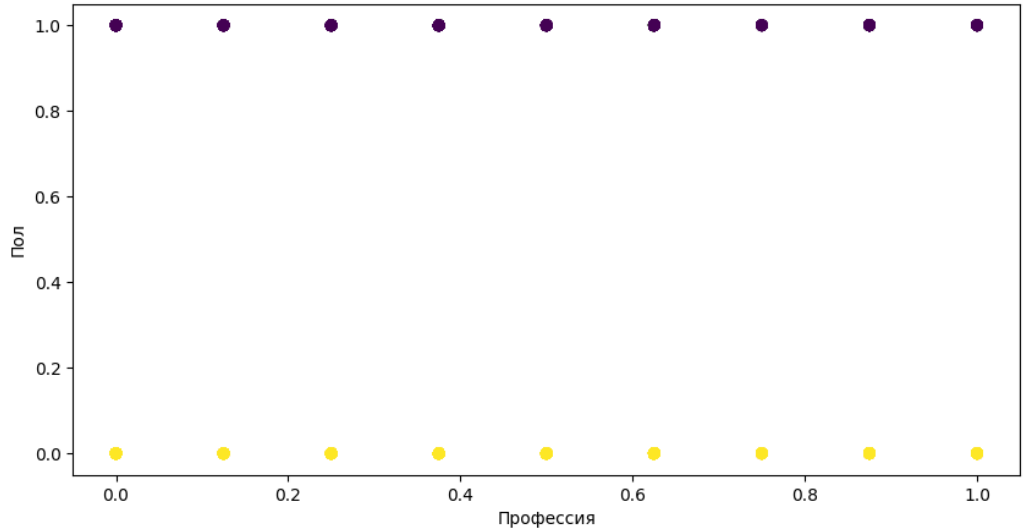

Рисунок 26. Количество кластеров 2

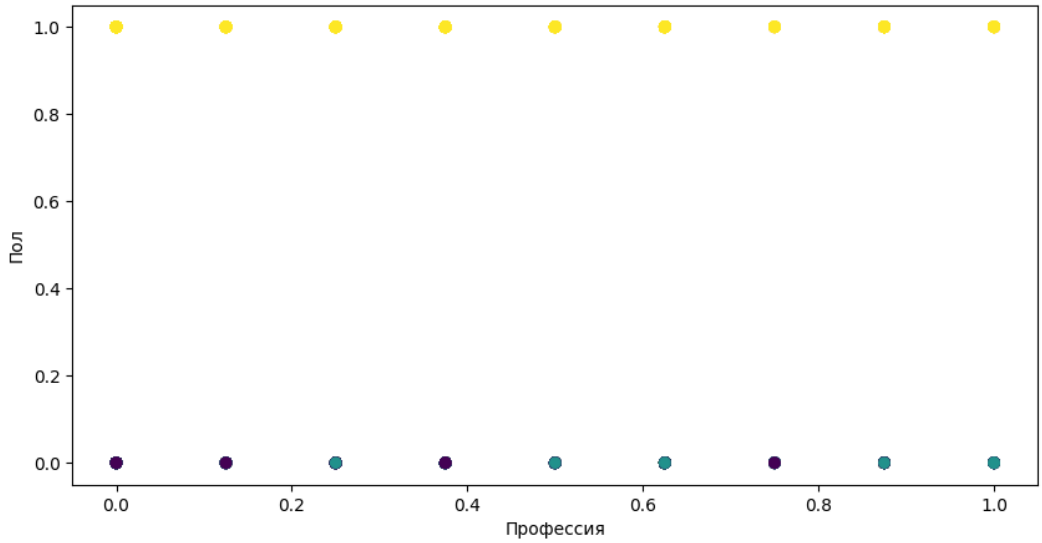

Рисунок 27. Количество кластеров 3

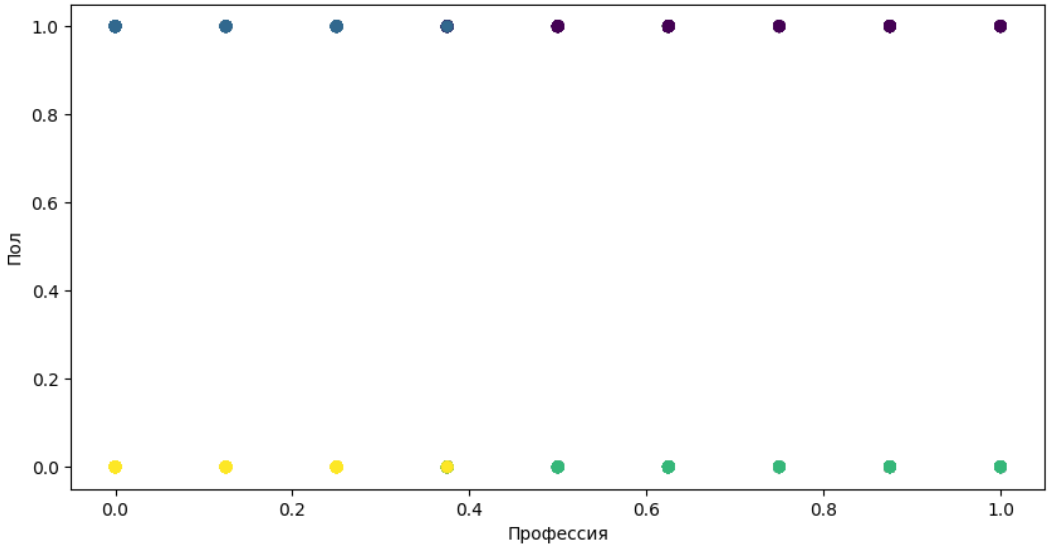

Рисунок 28. Количество кластеров 4

Вывод

Были проанализировать данные о клиентах магазина:

1. Выявлены зависимости по следующим факторным переменным:
   - **Gender** (Пол),
   - **Age** (Возраст),
   - **Spending Score (1-100)** (Оценка расходов (1-100)),
   - **Profession** (Профессия),
   - **Work Experience** (Опыт работы),
   - **Family Size** (Размер семьи),
   - **Annual Income (K$)** (Годовой доход (K$)).

2. Данные были разбиты на кластеры (рис. 14–17, 20-23, 26-28).

3. Оценено качество кластеризации для методов **K-means**, **SpectralClustering**, **BisectingKMeans** (рис. 13, 19, 25), и для всех методов максимальным размером силуэт (равен **0.33586090607117003**) был при количестве кластеров равным **2**.