

Сеть фитнес-центров «Культурист-датасаентист». Стратегия взаимодействия с клиентами на основе аналитических данных

Распространённая проблема фитнес-клубов и других сервисов — отток клиентов. Как понять, что клиент больше не с вами? Можно записать в отток тех, кто попросил закрыть договор или удалил аккаунт. Однако клиенты не всегда уходят демонстративно: чаще перестают пользоваться сервисом тихо.

Индикаторы оттока зависят от специфики отрасли. Когда пользователь редко, но стабильно закупается в интернет-магазине — не похоже, что он «отвалился». А вот если две недели не заходит на канал с ежедневно обновляемым контентом, дела плохи: подписчик заскучал и, кажется, оставил вас.

Для фитнес-центра можно считать, что клиент попал в отток, если за последний месяц ни разу не посетил спортзал. Конечно, не исключено, что он уехал на Бали и по приезде обязательно продолжит ходить на фитнес. Однако чаще бывает наоборот. Если клиент начал новую жизнь с понедельника, немного походил в спортзал, а потом пропал — скорее всего, он не вернётся. Чтобы бороться с оттоком, отдел по работе с клиентами «Культуриста-датасаентиста» перевёл в электронный вид множество клиентских анкет. Наша задача — провести анализ и подготовить план действий по удержанию клиентов. А именно:

- научиться прогнозировать вероятность оттока (на уровне следующего месяца) для каждого клиента;
- сформировать типичные портреты клиентов: выделить несколько наиболее ярких групп и охарактеризовать их основные свойства;
- проанализировать основные признаки, наиболее сильно влияющие на отток;
- сформулировать основные выводы и разработать рекомендации по повышению качества работы с клиентами:
 1. выделить целевые группы клиентов;
 2. предложить меры по снижению оттока;
 3. определить другие особенности взаимодействия с клиентами.

Описание данных

- 'Churn' — факт оттока в текущем месяце;

Текущие поля в датасете:

Данные клиента за предыдущий до проверки факта оттока месяц:

- 'gender' — пол;
- 'Near_Location' — проживание или работа в районе, где находится фитнес-центр;
- 'Partner' — сотрудник компании-партнёра клуба (сотрудничество с компаниями, чьи сотрудники могут получать скидки на абонемент — в таком случае фитнес-центр хранит информацию о работодателе клиента);

- 'Promo_friends' — факт первоначальной записи в рамках акции «приведи друга» (использовал промо-код от знакомого при оплате первого абонемента);
- 'Phone' — наличие контактного телефона;
- 'Age' — возраст;
- 'Lifetime' — время с момента первого обращения в фитнес-центр (в месяцах).

Информация на основе журнала посещений, покупок и информация о текущем статусе абонемента клиента:

- 'Contract_period' — длительность текущего действующего абонемента (месяц, 3 месяца, 6 месяцев, год);
- 'Month_to_end_contract' — срок до окончания текущего действующего абонемента (в месяцах);
- 'Group_visits' — факт посещения групповых занятий;
- 'Avg_class_frequency_total' — средняя частота посещений в неделю за все время с начала действия абонемента;
- 'Avg_class_frequency_current_month' — средняя частота посещений в неделю за предыдущий месяц;
- 'Avg_additional_charges_total' — суммарная выручка от других услуг фитнес-центра: кафе, спорт-товары, косметический и массажный салон.

План работы - содержание

1. Открытие файла. Изучение общей информации
2. Исследовательский анализ данных
3. Модель прогнозирования оттока клиентов
4. Кластеризация клиентов
5. Выводы и базовые рекомендации

Открытие файла. Изучение общей информации

```
In [1]: import pandas as pd
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
import matplotlib.pyplot as plt
from sklearn.metrics import accuracy_score, precision_score, recall_score
from scipy.cluster.hierarchy import dendrogram, linkage
from sklearn.cluster import KMeans
```

```
In [2]: gym_churn = pd.read_csv('datasets/gym_churn.csv')
```

```
In [3]: display(gym_churn.head())
```

	gender	Near_Location	Partner	Promo_friends	Phone	Contract_period	Group_visits	Age	Avg.
0	1	1	1	1	0	6	1	29	
1	0	1	0	0	1	12	1	31	
2	0	1	1	0	1	1	0	28	

	gender	Near_Location	Partner	Promo_friends	Phone	Contract_period	Group_visits	Age	Avg.
3	0	1	1	1	1	12	1	33	
4	1	1	1	1	1	1	0	26	

In [4]: `print(gym_churn.info())`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4000 entries, 0 to 3999
Data columns (total 14 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   gender          4000 non-null    int64  
 1   Near_Location   4000 non-null    int64  
 2   Partner         4000 non-null    int64  
 3   Promo_friends  4000 non-null    int64  
 4   Phone           4000 non-null    int64  
 5   Contract_period 4000 non-null    int64  
 6   Group_visits   4000 non-null    int64  
 7   Age             4000 non-null    int64  
 8   Avg_additional_charges_total 4000 non-null    float64 
 9   Month_to_end_contract 4000 non-null    float64 
 10  Lifetime        4000 non-null    int64  
 11  Avg_class_frequency_total 4000 non-null    float64 
 12  Avg_class_frequency_current_month 4000 non-null    float64 
 13  Churn           4000 non-null    int64  
dtypes: float64(4), int64(10)
memory usage: 437.6 KB
None
```

В датасете 4000 строк, пропущенных значений нет. Все значения числовые

Перейдем к исследовательскому анализу

Исследовательский анализ данных

In [5]: `gym_churn.describe()`

	gender	Near_Location	Partner	Promo_friends	Phone	Contract_period	Grou
count	4000.000000	4000.000000	4000.000000	4000.000000	4000.000000	4000.000000	4000.
mean	0.510250	0.845250	0.486750	0.308500	0.903500	4.681250	0.
std	0.499957	0.361711	0.499887	0.461932	0.295313	4.549706	0.
min	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	0.
25%	0.000000	1.000000	0.000000	0.000000	1.000000	1.000000	0.
50%	1.000000	1.000000	0.000000	0.000000	1.000000	1.000000	0.
75%	1.000000	1.000000	1.000000	1.000000	1.000000	6.000000	1.
max	1.000000	1.000000	1.000000	1.000000	1.000000	12.000000	1.

Что мы видим сразу -

- мужчин и женщин почти поровну
- люди чаще ходят в зал, расположенный рядом с домом или работой

- партнерских абонементов почти столько же, сколько и непартнерских
- акция "приведи" друга не очень популярна (среднее значение 0,3)
- телефон оставили почти все посетители
- средний абонемент - 4,68 месяца
- групповые занятия посещают меньше половины клиентов
- средний возраст клиента - 30 лет
- клиенты в среднем тратят +147 единиц на доп услуги
- в основном в датасете "новички" - среднее время с момента первого обращения в фитнес-клуб - 3,7 месяца
- в среднем люди посещают фитнес клуб около 2x раз в неделю

Нереально больших значений признаков нет

Разобьем датасет на оставшихся и "отвалившихся" клиентов

```
In [6]: gym_churn.groupby('Churn').mean()
```

	gender	Near_Location	Partner	Promo_friends	Phone	Contract_period	Group_visits
Churn							
0	0.510037	0.873086	0.534195	0.353522	0.903709	5.747193	0.464103 2
1	0.510839	0.768143	0.355325	0.183789	0.902922	1.728558	0.268615 2

Сразу бросается в глаза -

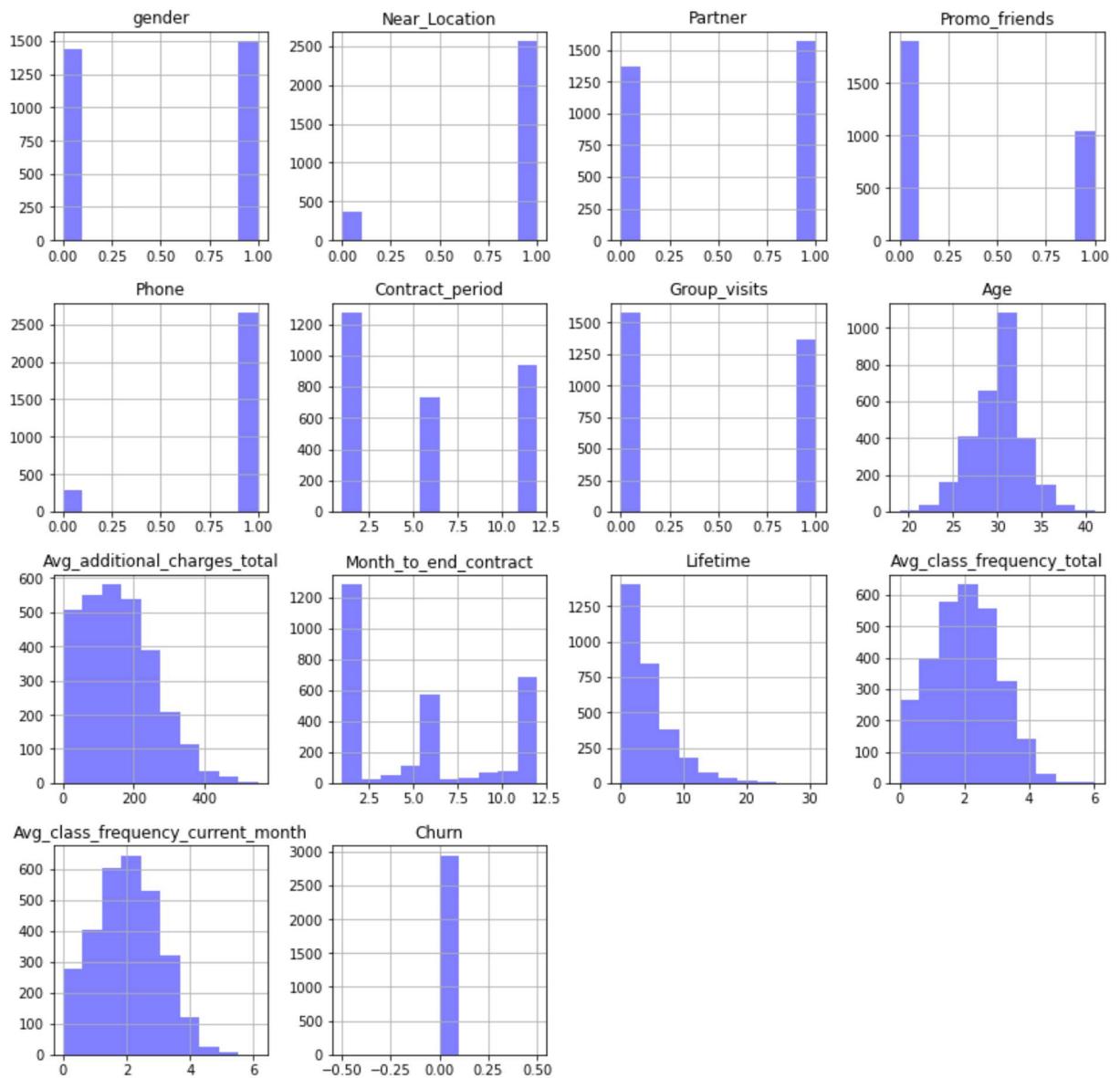
- ушедшие клиенты чаще приходе не по рекомендации
- брали "короткий" абонемент (меньше 2x месяцев)
- групповые занятия не посещают
- меньше тратят на доп услуги
- моложе тех, кто решил остаться
- до конца абонемента осталось меньше 2ч месяцев
- пришли в фитнес-клуб недавно (меньше месяца назад)
- ходят 1, 5 раза в неделю в среднем, в последний месяц - 1 раз в неделю

Подтвердим эти выводы визуализацией

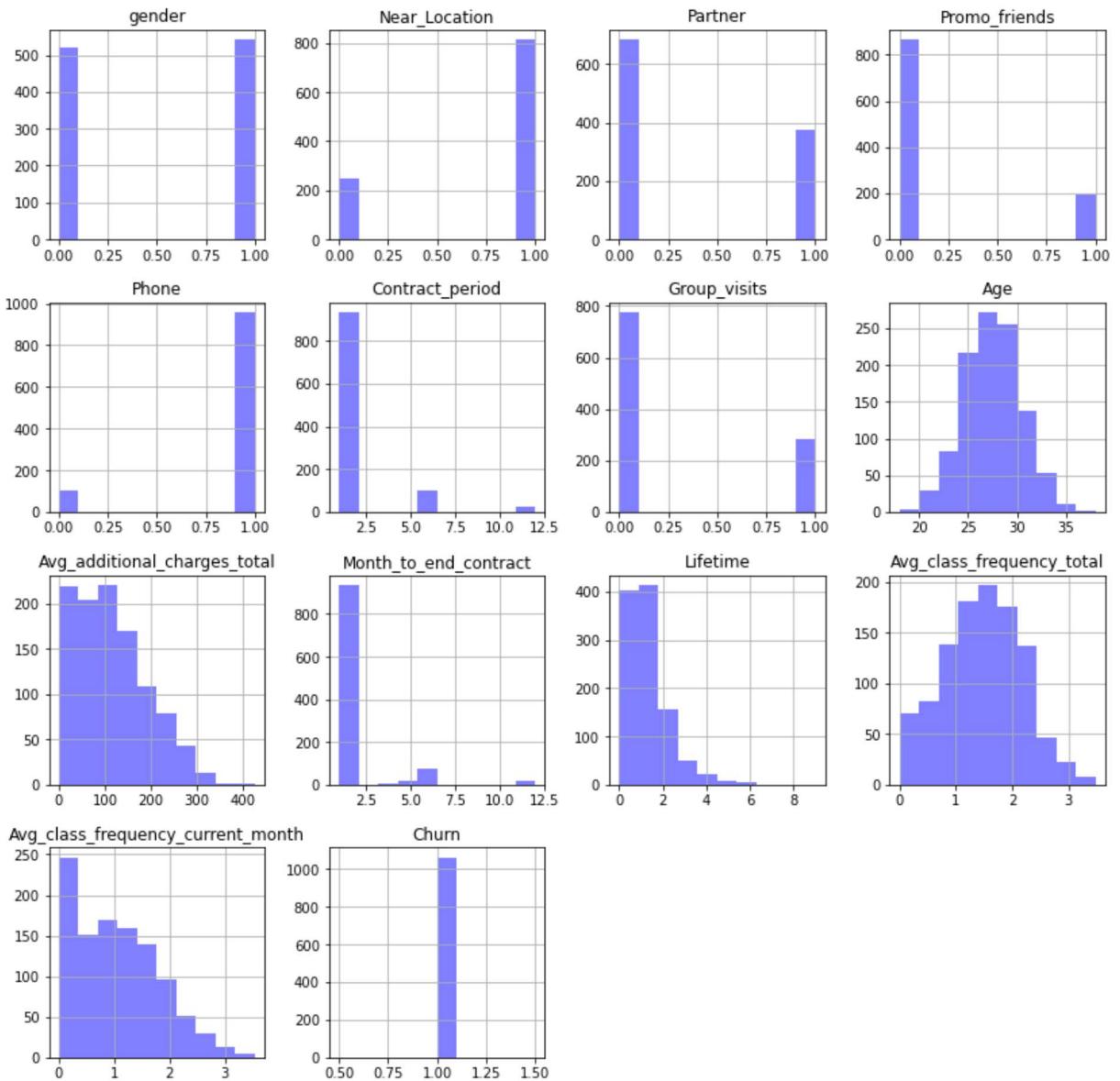
```
In [7]: for i in range (0, 2):
    if i==0:
        print('Распределение оставшихся клиентов по признакам')
    else:
        print('Распределение оттоковых клиентов по признакам')

    gym_churn.loc[gym_churn['Churn'] == i].hist(color = 'b', alpha = 0.5, figsize =
plt.show()
```

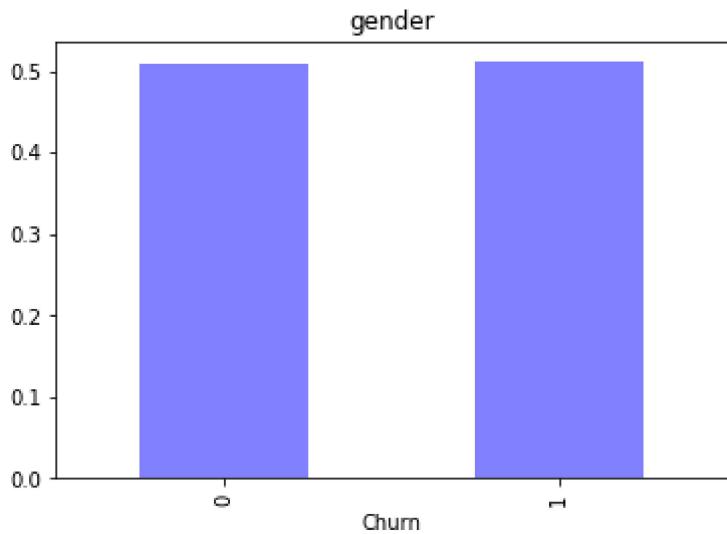
Распределение оставшихся клиентов по признакам

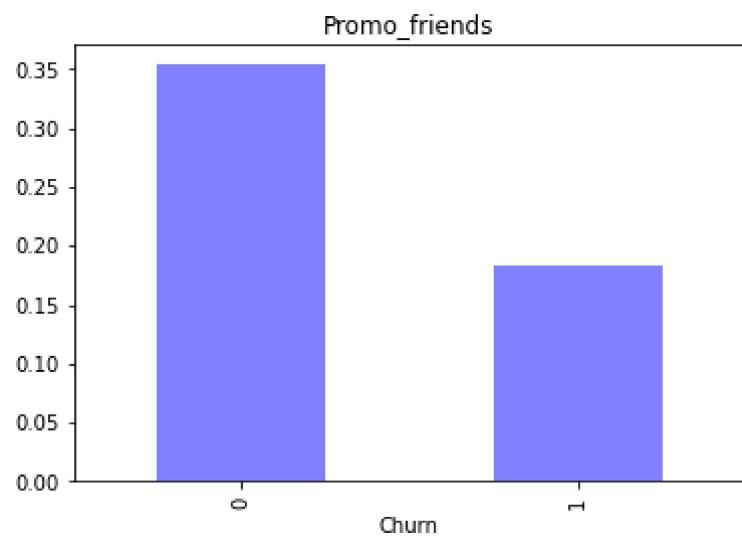
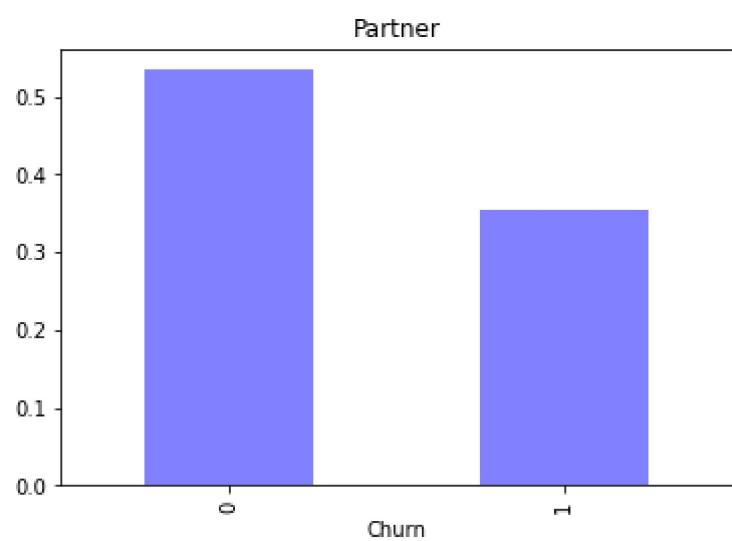
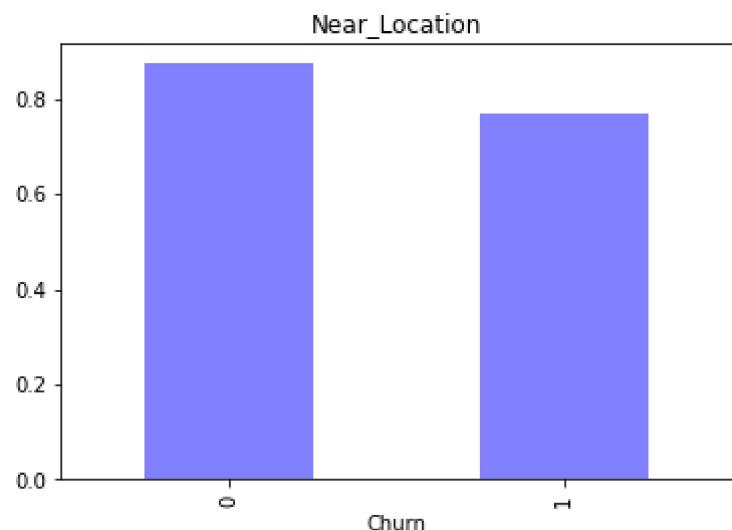


Распределение оттоковых клиентов по признакам

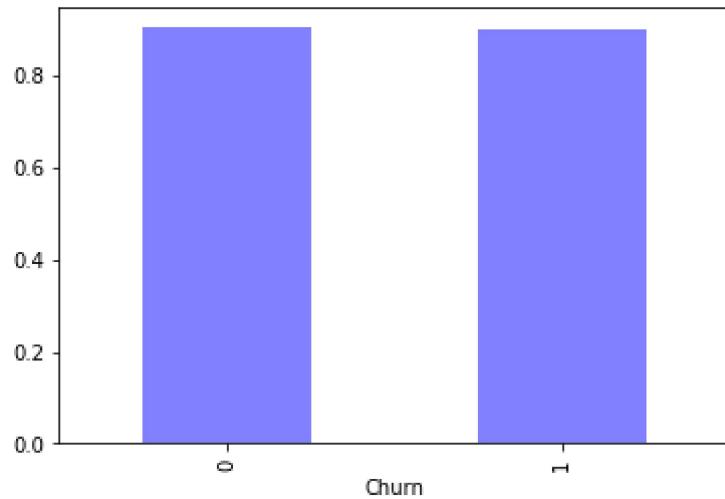


```
In [8]: for col in gym_churn.groupby('Churn').mean().columns:
    gym_churn.groupby('Churn').mean()[col].plot(kind = 'bar', color = 'b', alpha = 0.5)
    plt.title(col)
    plt.show()
```

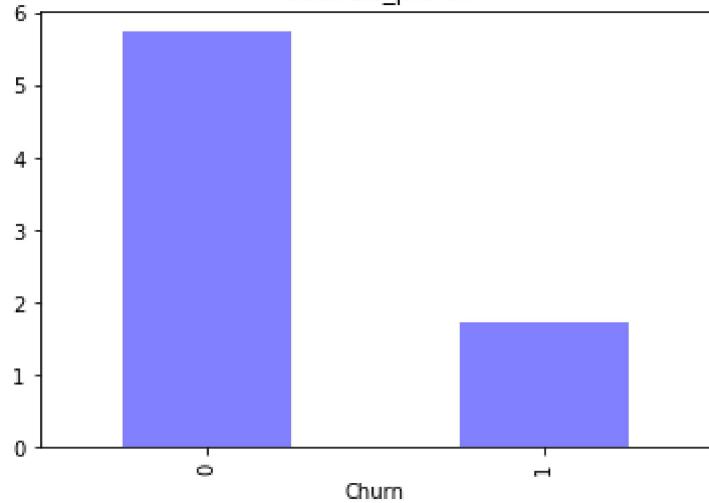




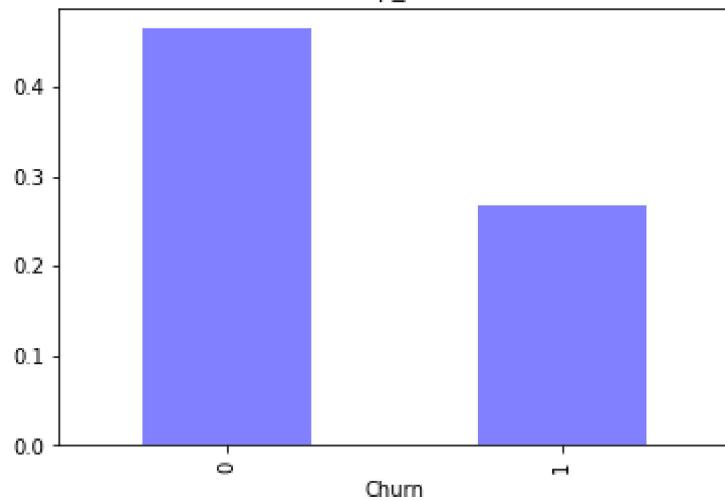
Phone

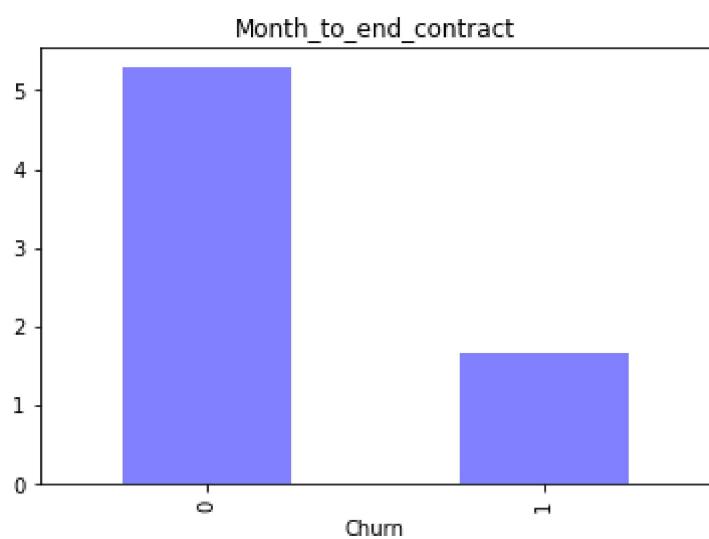
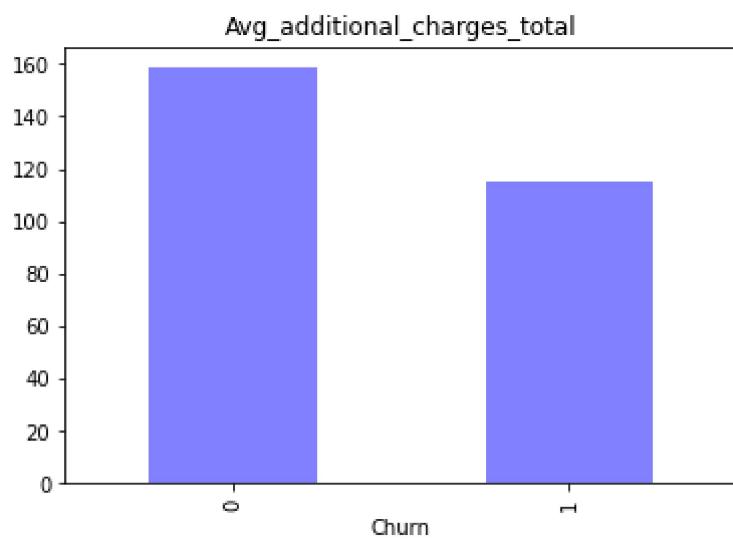
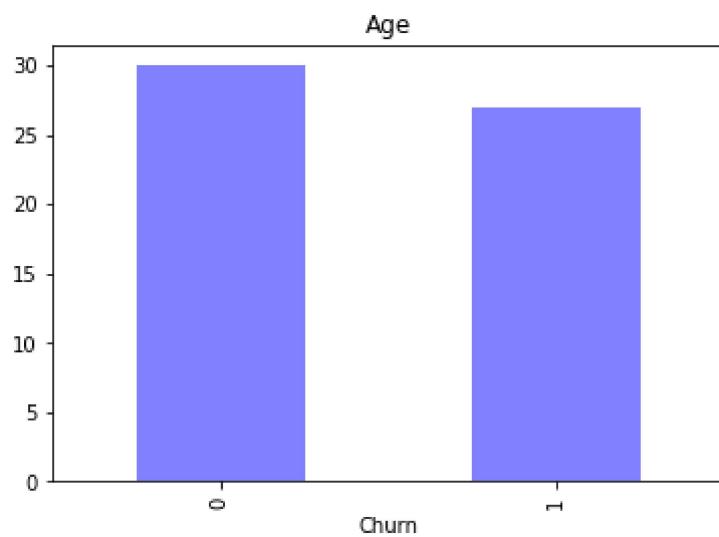


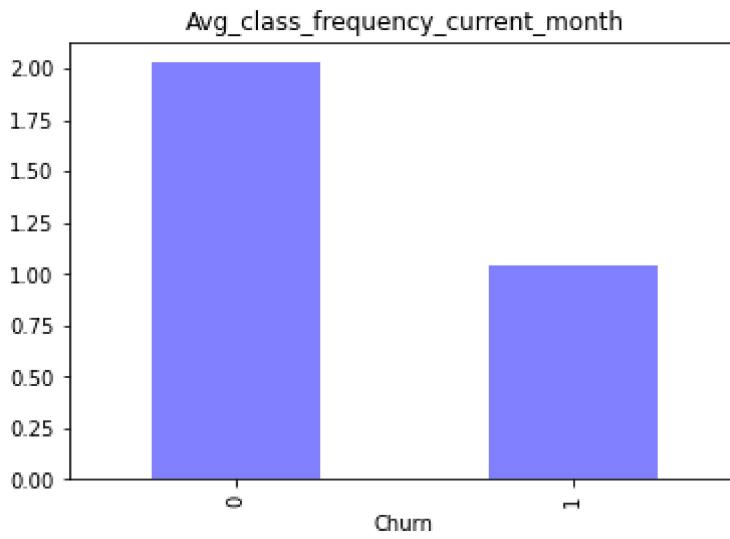
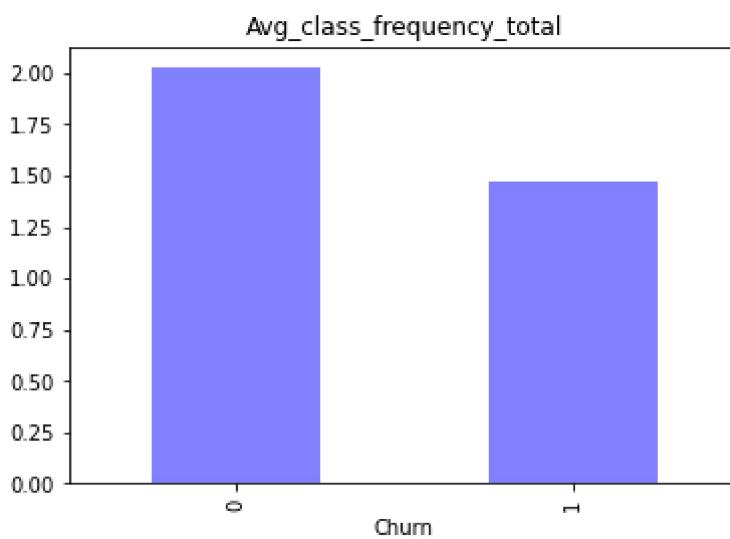
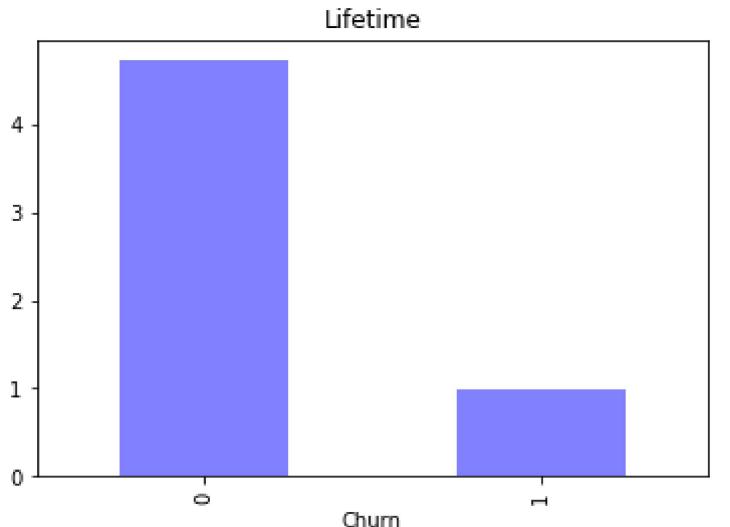
Contract_period



Group_visits







Постоим матрицу коррелий и найдем явные и сильные зависимости с переменной оттока (Churn)

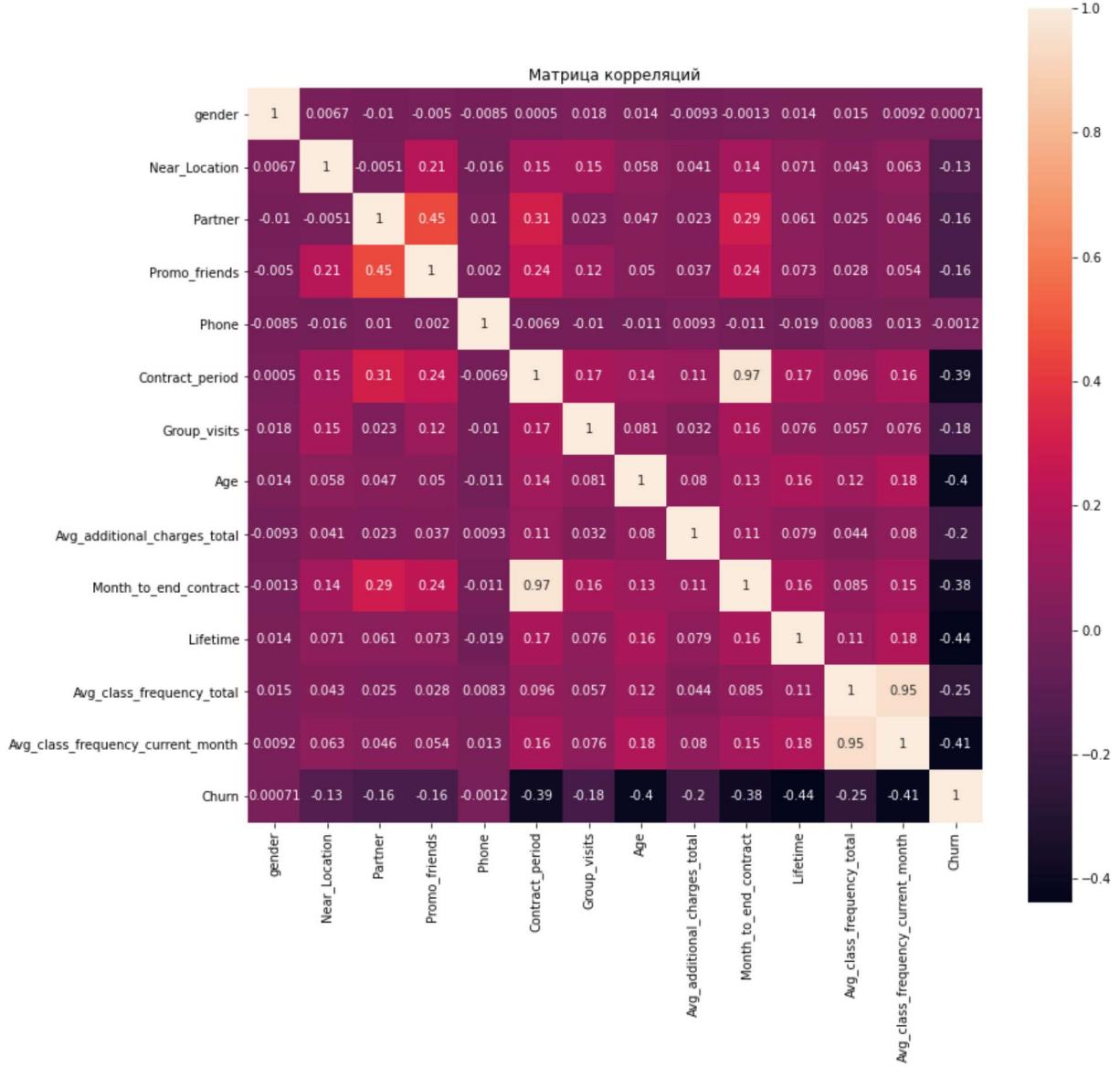
In [9]: `gym_churn.corr()`

Out[9]:

	gender	Near_Location	Partner	Promo_friends	Phone	C
gender	1.000000	0.006699	-0.010463	-0.005033	-0.008542	
Near_Location	0.006699	1.000000	-0.005119	0.210964	-0.015763	
Partner	-0.010463	-0.005119	1.000000	0.451960	0.009970	

	gender	Near_Location	Partner	Promo_friends	Phone	C
Promo_friends	-0.005033	0.210964	0.451960	1.000000	0.001982	
Phone	-0.008542	-0.015763	0.009970	0.001982	1.000000	
Contract_period	0.000502	0.150233	0.306166	0.244552	-0.006893	
Group_visits	0.017879	0.154728	0.022710	0.120170	-0.010099	
Age	0.013807	0.058358	0.047480	0.050113	-0.011403	
Avg_additional_charges_total	-0.009334	0.040761	0.022941	0.036898	0.009279	
Month_to_end_contract	-0.001281	0.143961	0.294632	0.239553	-0.011196	
Lifetime	0.013579	0.070921	0.061229	0.072721	-0.018801	
Avg_class_frequency_total	0.014620	0.043127	0.024938	0.028063	0.008340	
Avg_class_frequency_current_month	0.009156	0.062664	0.045561	0.053768	0.013375	
Churn	0.000708	-0.128098	-0.157986	-0.162233	-0.001177	

```
In [10]: plt.figure(figsize=(13, 13))
sns.heatmap(gym_churn.corr(), annot=True, square=True)
plt.title('Матрица корреляций')
plt.show()
```



С целевой переменной Churn нет заметно сильных корреляций, зато есть признаки, который совсем не коррелируют с оттоком - пол и номер телефона

Будем смотреть дальше. На этих данных можно построить модель

Модель прогнозирования оттока клиентов

Разделим данные на признаки (матрица X) и целевую переменную (y)

```
In [11]: X = gym_churn.drop('Churn', axis =1)
y = gym_churn['Churn']
```

Разделим модель на обучающую и валидационную выборки

```
In [12]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Обучим модель на train-выборке двумя способами:

- логистической регрессией,
- случайным лесом.

```
In [13]: lg_model = LogisticRegression(max_iter = 1000)
lg_model.fit(X_train, y_train)
lg_predictions = lg_model.predict(X_test)
```

```
In [14]: rf_model = RandomForestClassifier()
rf_model.fit(X_train, y_train)
rf_predictions = rf_model.predict(X_test)
```

Оценим метрики accuracy, precision и recall для обеих моделей на валидационной выборке

```
In [15]: def metrics(y_true, y_pred, title = 'Метрики классификации'):
    print(title)
    print('\tAccuracy: {:.2f}'.format(accuracy_score(y_true, y_pred)))
    print('\tPrecision: {:.2f}'.format(precision_score(y_true, y_pred)))
    print('\tRecall: {:.2f}'.format(recall_score(y_true, y_pred)))
```

```
In [16]: metrics(y_test, lg_predictions)
```

Метрики классификации
 Accuracy: 0.93
 Precision: 0.89
 Recall: 0.85

```
In [17]: metrics(y_test, rf_predictions)
```

Метрики классификации
 Accuracy: 0.92
 Precision: 0.87
 Recall: 0.82

По метрикам выигрывает логическая регрессия

Кластеризация клиентов

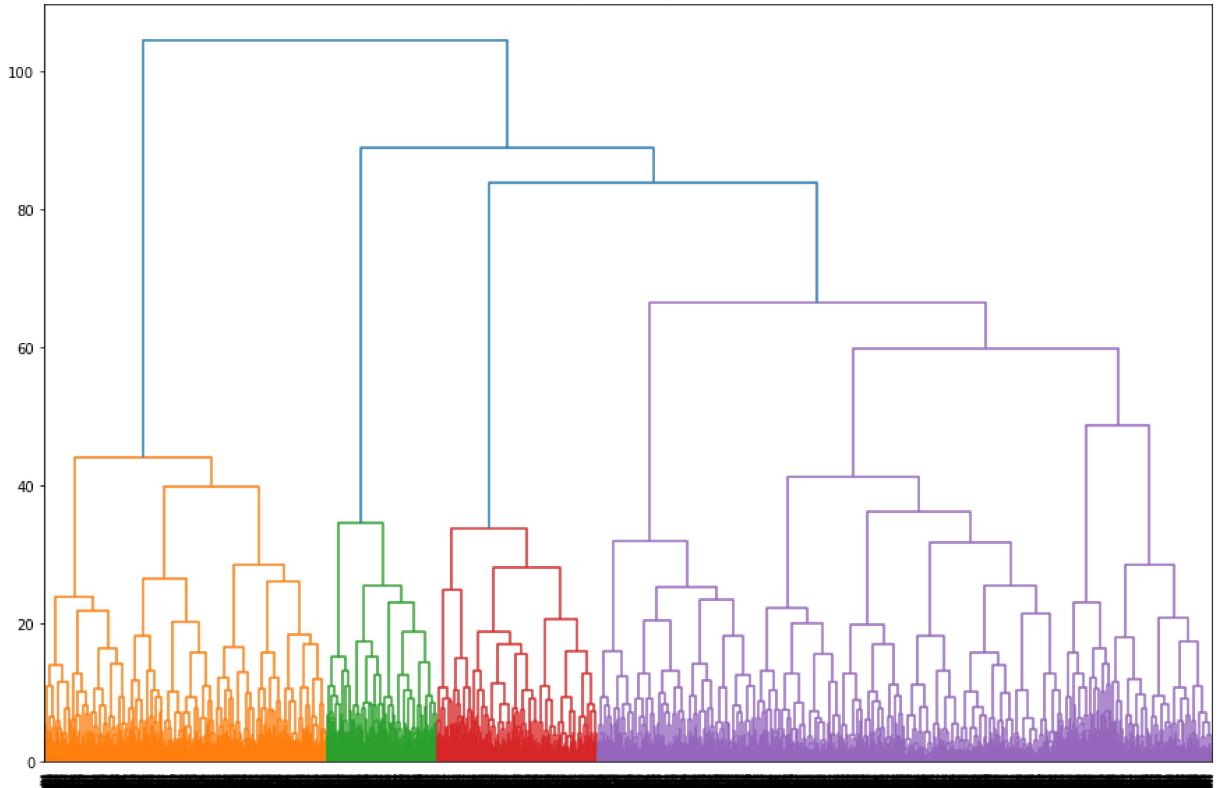
Стандартизуем данные, построим матрицу расстояний и нарисуем по ней дендрограмму

```
In [18]: scaler = StandardScaler()
X_sc = scaler.fit_transform(X)
```

```
In [19]: linked = linkage(X_sc, method='ward')
```

```
In [20]: plt.figure(figsize=(15, 10))
dendrogram(linked, orientation='top')
plt.title('Кластеризация признаков')
plt.show()
```

Кластеризация признаков



На основе графика можно выделить 4 кластера, на основе задания - 5 :)

4 кластер получился, как 3 первых, поэтому разобьем клиентов на 5 кластеров

```
In [21]: km = KMeans(n_clusters=5)
```

```
In [22]: labels = km.fit_predict(X_sc)
```

```
In [23]: gym_churn['clusters_km'] = labels
```

```
In [24]: display(gym_churn.groupby('clusters_km').mean())
```

	gender	Near_Location	Partner	Promo_friends	Phone	Contract_period	Group_visits
clusters_km							
0	0.497041	0.000000	0.461538	0.078895	1.0	2.378698	0.218935
1	0.483428	1.000000	0.355699	0.242522	1.0	1.959580	0.340340
2	0.523316	0.862694	0.471503	0.305699	0.0	4.777202	0.427461
3	0.498531	0.960823	0.783546	0.575906	1.0	10.854065	0.539667
4	0.565371	0.977621	0.342756	0.221437	1.0	2.554770	0.472320

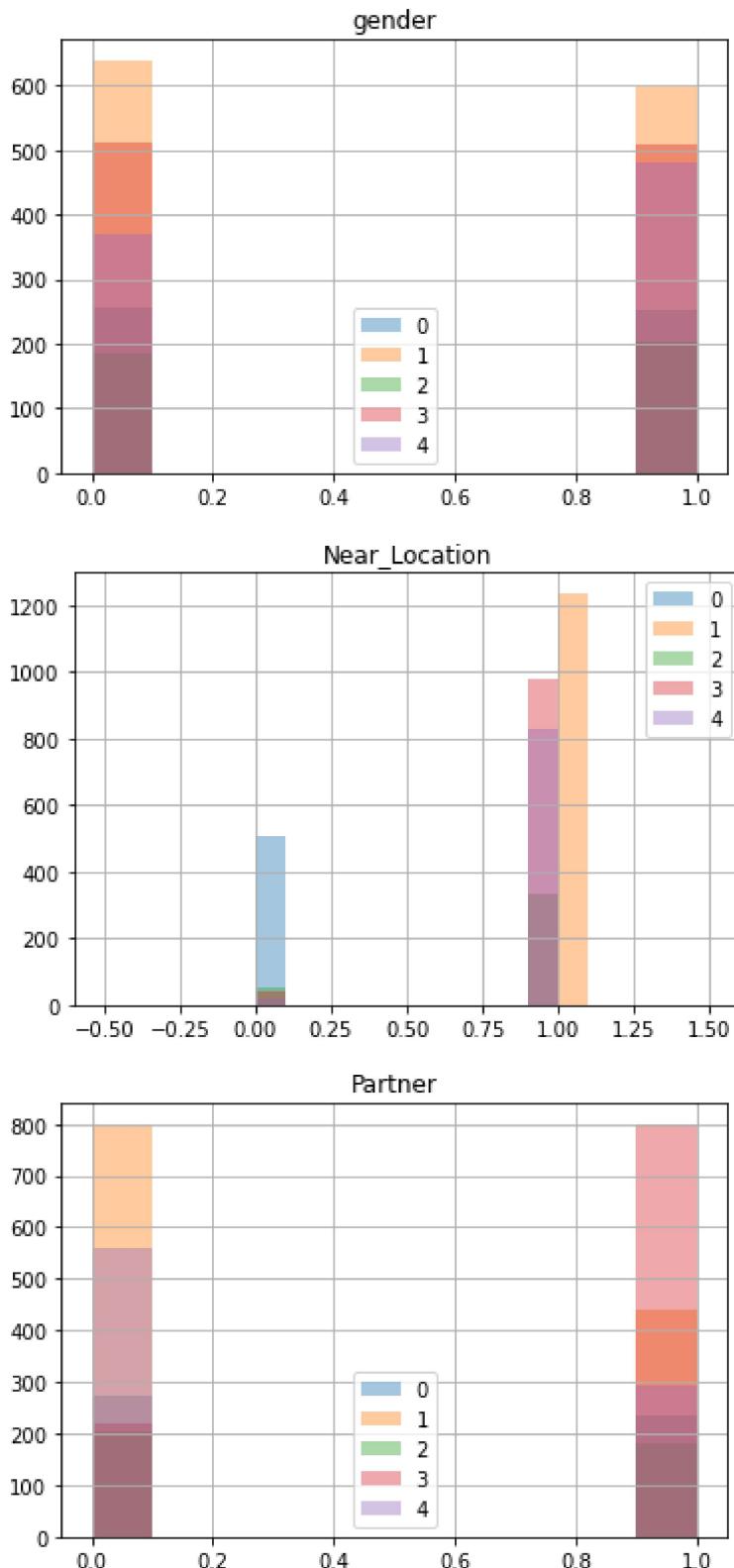
По средним значения можно сразу выделить, что у кластера клиентов №0 не указан номер телефона, кластер 3 - кластер долгих абонементов (средняя длительность 10,88 месяцев), кластер 2 - содержит клиентов, которые чаще остаются в клубе

ПОсмотрим на распределение по признакам

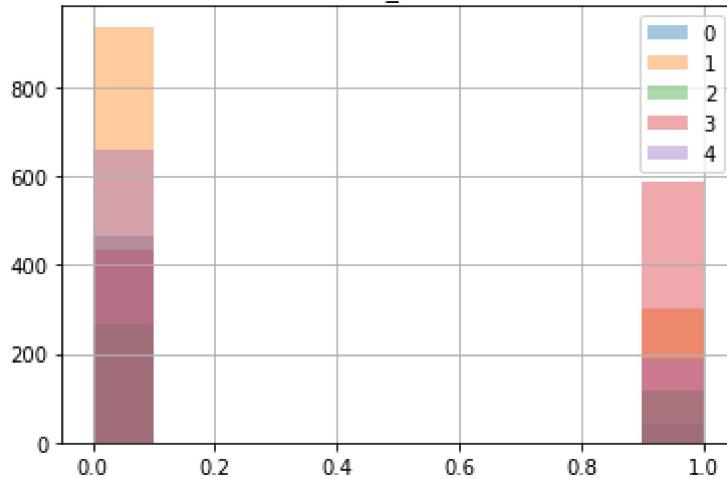
```
In [25]: for col in gym_churn.columns:
    for i in range (0, len(gym_churn['clusters_km'].unique())):
```

```
gym_churn.loc[gym_churn['clusters_km']==i][col].hist(alpha=0.4)
plt.title(col)

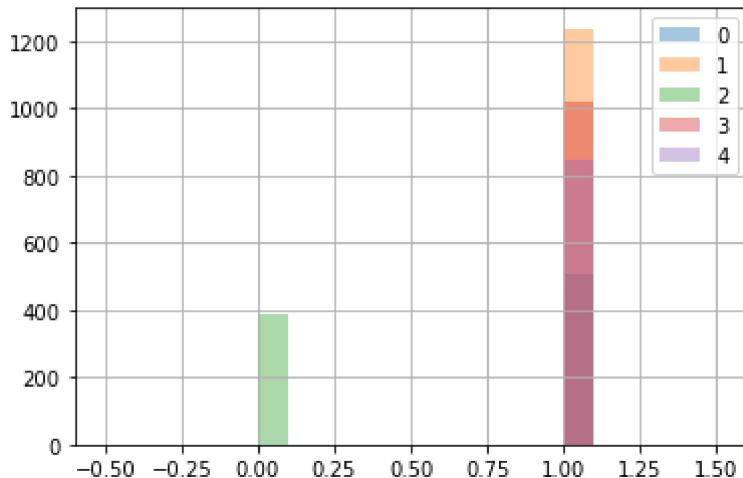
plt.legend(labels=('0', '1', '2', '3', '4'))
plt.show()
```



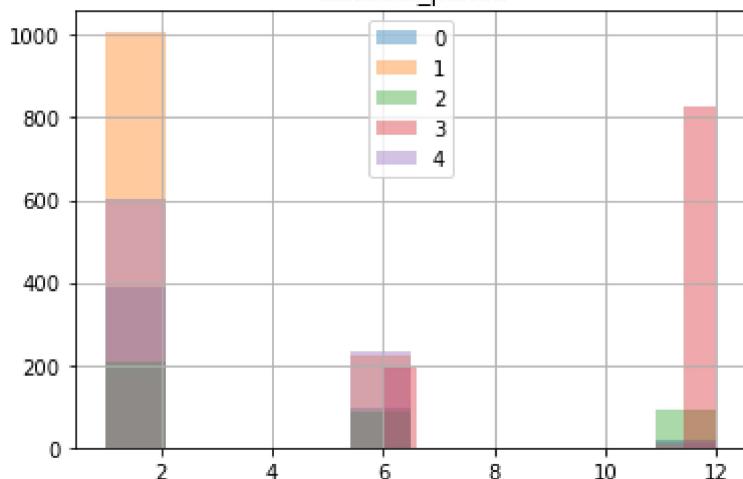
Promo_friends

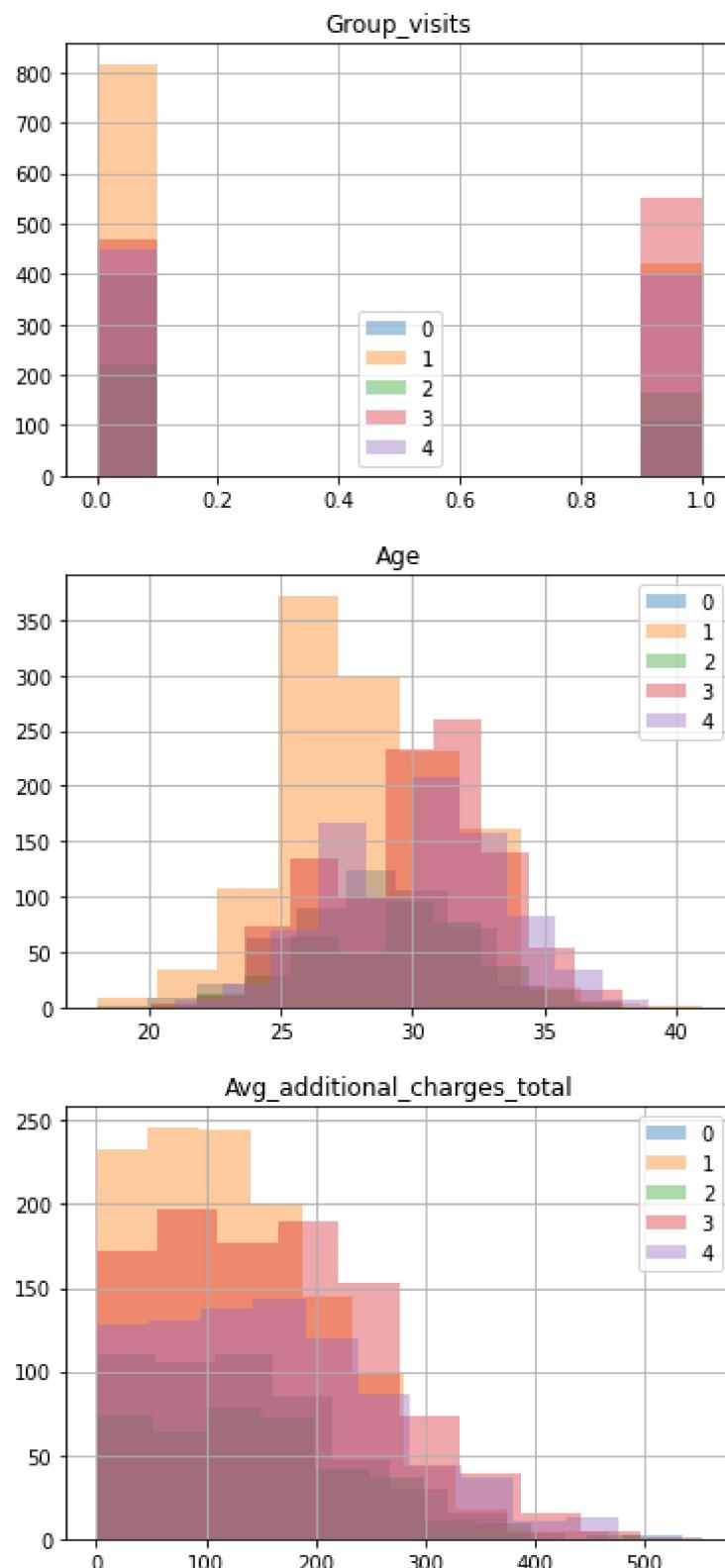


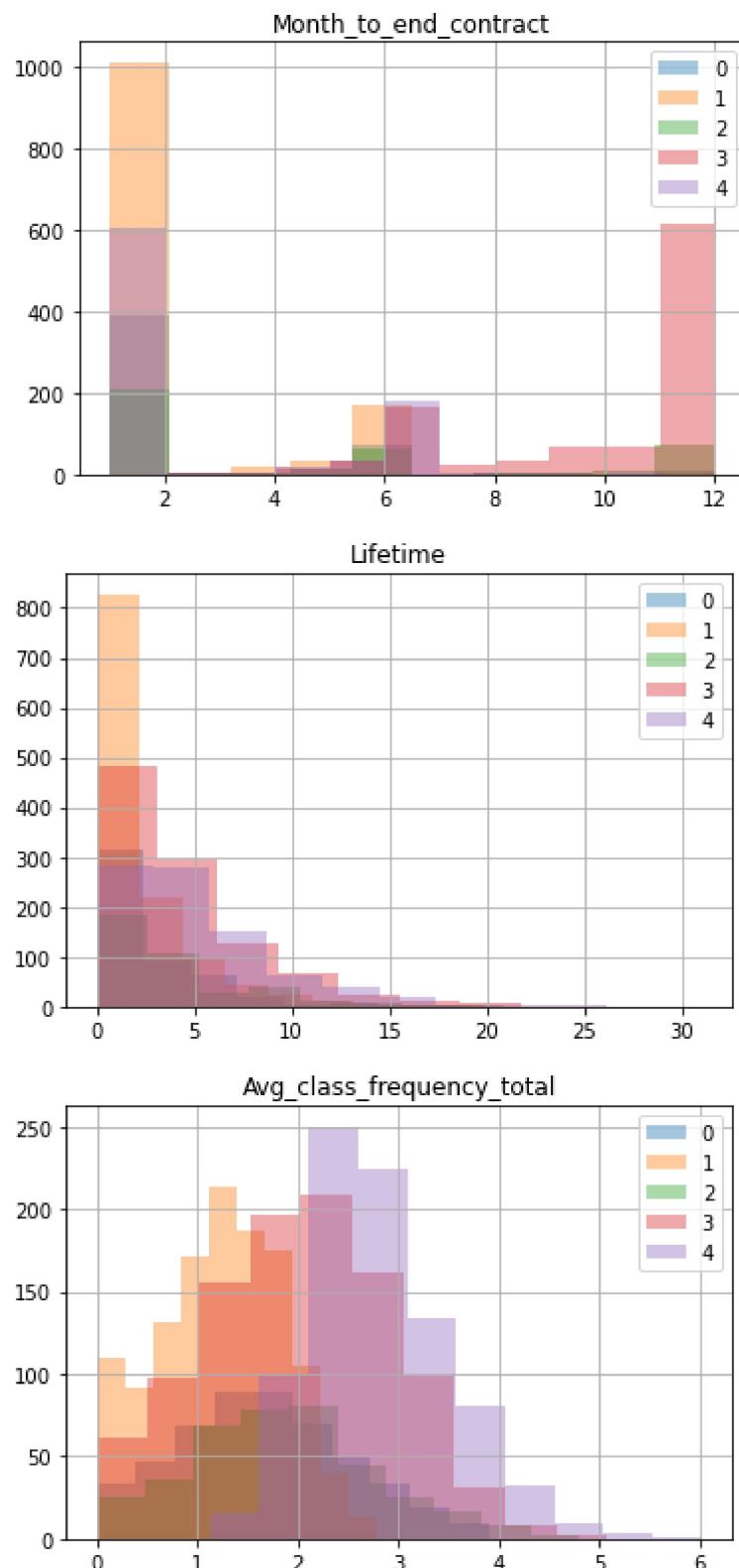
Phone

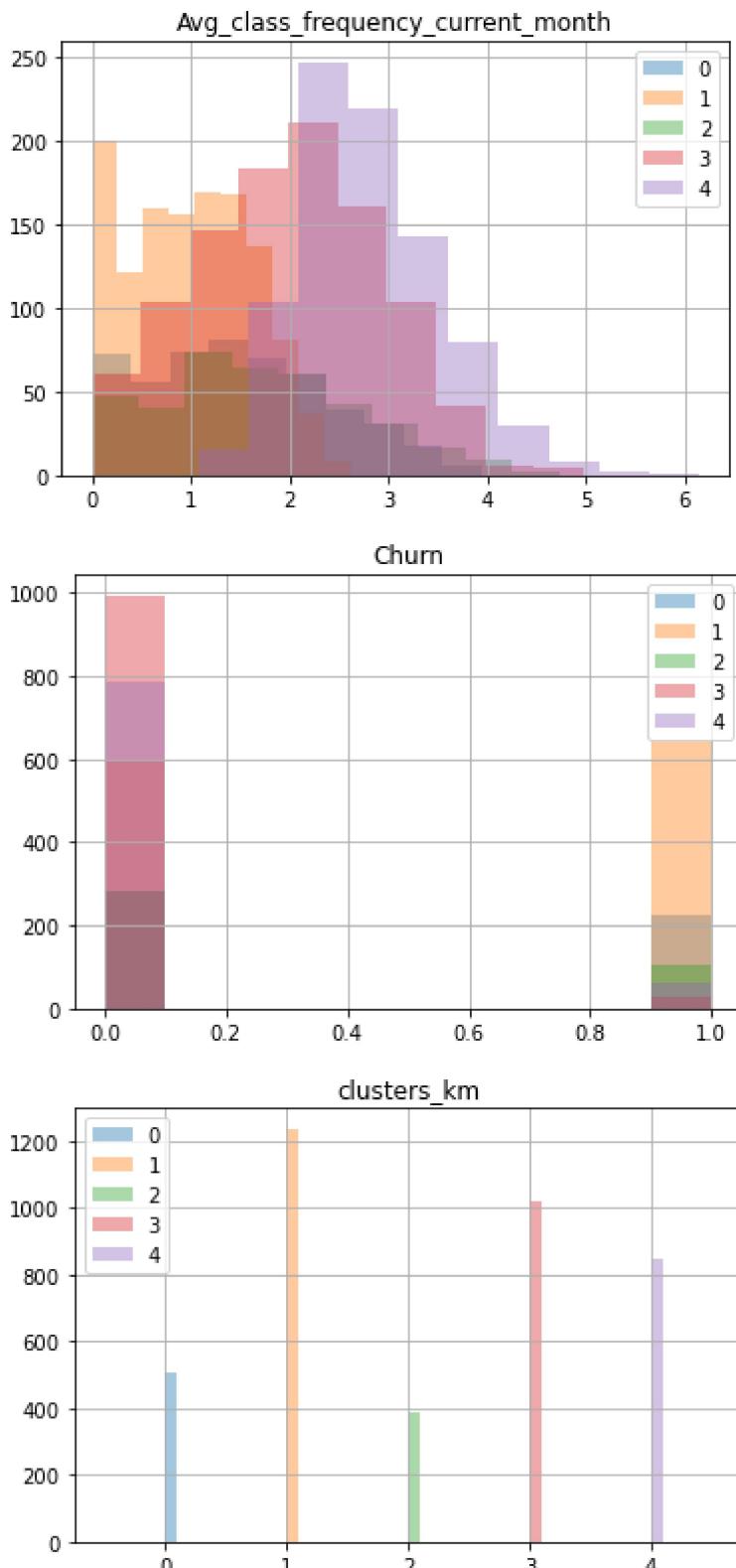


Contract_period









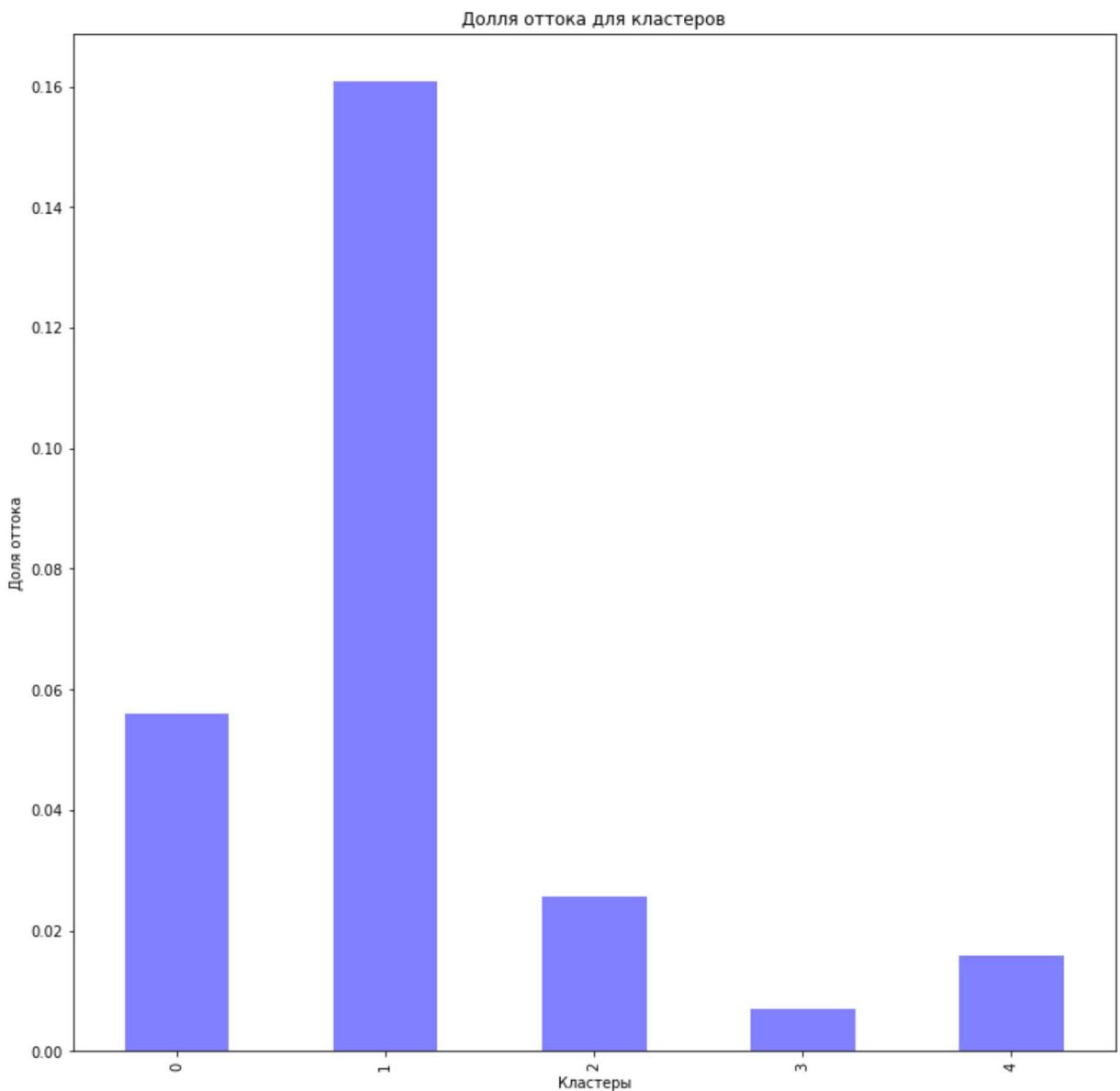
- пол распределен равномерно
- кластер 1 живет/работает далеко от клуба
- кластер 3 в основном партнерские клиенты
- кластер 4 не привели друзья
- кластер 0 не оставил номер телефона
- кластер 3 с долгими годовыми абонементами
- кластер 4 пользуется меньше групповыми занятиями
- кластер 4 моложе
- все кластеры равно тратят средств на доп услуги
- кластер 2 чаще посещает клуб

- кластер 3 чаще остается в клубе

```
In [26]: display(gym_churn.groupby('clusters_km')['Churn'].sum()/4000)

clusters_km
0    0.05600
1    0.16075
2    0.02575
3    0.00700
4    0.01575
Name: Churn, dtype: float64
```

```
In [27]: plt.figure(figsize=(13,13))
(gym_churn.groupby('clusters_km')['Churn'].sum()/4000).plot(kind = 'bar', color = 'blue')
plt.title('Доля оттока для кластеров')
plt.xlabel('Кластеры')
plt.ylabel('Доля оттока')
plt.show()
```



Кластер 4 - чемпион по оттоку клиентов. Из ранних выводов - его не привели друзья, он не посещает групповые программы и он моложе

Выводы и базовые рекомендации

Клиент, которые остается:

- живет рядом
- берет абонемент на длительный срок
- посещает групповые занятия
- приобретает доп услуги
- не из партнерской программы

Что можно улучшить, чтобы удержать клиентов

Так как клиенты, которые отваливаются, моложе и не посещают групповые занятия, можно сделать групповые "молодежные модные" групповые программы. Также известно, что их чаще не привели друзья. Возможно, это одиночки, которым некому подсказать, что делать в зале, нужно усилить тренерскую работу, либо запустить акцию для клиентов "помоги другому освоиться"

In []: