Прогнозирование вероятности оттока пользователей для фитнес-центров

Содержание

- 1 Описание проекта
- 2 Загрузите данные
  - 2.1 Подготовка данных
    - 2.1.1 Замена названий столбцов
    - 2.1.2 Проверка на наличия дублей и пропусков
- 3 исследовательский анализ данных (EDA)
  - 3.1 Посмотрим на датасет: есть ли в нем отсутствующие признаки, изучите средние значения и стандартные отклонения
  - 3.2 Посмотрите на средние значения признаков в двух группах — тех, кто ушел в отток и тех, кто остался
  - 3.3 Постройте столбчатые гистограммы и распределения признаков для тех, кто ушёл (отток) и тех, кто остался (не попали в отток)
  - 3.4 Постройте матрицу корреляций и отобразите её.
- 4 Постройте модель прогнозирования оттока клиентов
  - 4.1 Разбейте данные на обучающую и валидационную выборку функцией
  - 4.2 Обучите модель на train-выборке двумя способами: логистической регрессией, случайным лесом.
  - 4.3 Оцените метрики accuracy, precision и recall для обеих моделей на валидационной выборке. Сравните по ним модели. Какая модель показала себя лучше на основании метрик?
- 5 Кластеризацию клиентов
  - 5.1 Стандартизируйте данные
  - 5.2 Постройте матрицу расстояний функцией linkage() на стандартизованной матрице признаков и нарисуйте дендрограмму. На основании полученного графика предположите, какое количество кластеров можно выделить.
  - 5.3 Обучите модель кластеризации на основании алгоритма K-Means и спрогнозируйте кластеры клиентов.
  - 5.4 Посмотрите на средние значения признаков для кластеров. Можно ли сразу что-то заметить?
  - 5.5 Постройте распределения признаков для кластеров. Можно ли что-то заметить по ним?
  - 5.6 Для каждого полученного кластера посчитайте долю оттока (методом groupby()). Отличаются ли они по доле оттока? Какие кластеры склонны к оттоку, а какие — надёжны?
- 6 Общий вывод

Описание проекта

Сеть фитнес-центров «Культурист-датасаентист» разрабатывает стратегию взаимодействия с клиентами на основе аналитических данных. Распространённая проблема фитнес-клубов и других сервисов — отток клиентов. Как понять, что клиент больше не с вами? 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.
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.

Загрузите данные

Заказчик подготовил данные, которые содержат данные на месяц до оттока и факт оттока на определённый месяц.
Набор данных включает следующие поля:

- `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, 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, спорттовары, косметический и массажный салон.

`Churn` - факт оттока в текущем месяце.

In [1]:
```
# импорт библиотек
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

#ML
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split

from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1

from sklearn.tree import DecisionTreeRegressor, DecisionTreeClassifier
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifi
from sklearn.metrics import accuracy_score, precision_score, recall_score
from sklearn.cluster import KMeans
from scipy.cluster.hierarchy import dendrogram, linkage

# дополнительные настройки
import warnings

warnings.simplefilter('ignore')
```

In [2]:
```
path = 'datasets/gym_churn.csv'
df = pd.read_csv(path)
df.sample(5)
```

Out[2]:

	gender	Near_Location	Partner	Promo_friends	Phone	Contract_period	Group_visits	Age
146	1	1	0	0	1	1	0	32
177	1	1	0	1	1	1	1	27
3819	0	1	1	1	1	12	0	33
3870	0	1	0	0	0	1	1	24
1095	1	1	1	1	1	6	1	30

Подготовка данных

Замена названий столбцов

В названиях присутствуют заглавные буквы. Приведем их к нижнему регистру

In [3]:
```
df.columns = df.columns.str.lower()
```

```
df.columns
```

Out[3]:
```
Index(['gender', 'near_location', 'partner', 'promo_friends', 'phone',
       'contract_period', 'group_visits', 'age',
       'avg_additional_charges_total', 'month_to_end_contract', 'lifetime',
       'avg_class_frequency_total', 'avg_class_frequency_current_month',
       'churn'],
      dtype='object')
```

Проверка на наличия дублей и пропусков

In [4]:
```
print('Количество дублей {}'. format(df.duplicated().sum()))
```

```
Количество дублей 0
```

In [5]:
```
df.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
```

Вывод Как видно из приведенной выше таблице предобработка не требуется. Дублей и пропусков нет. Для удобства проведения дальнейшего исследования названия всех столбцоав приведены к нижнему регистру.

исследовательский анализ данных (EDA)

Посмотрим на датасет: есть ли в нем отсутствующие признаки, изучите средние значения и стандартные отклонения

In [6]:
```
df.describe().T
```

Out[6]:

	count	mean	std	min	25%	5(
gender	4000.0	0.510250	0.499957	0.000000	0.000000	1.000(
near_location	4000.0	0.845250	0.361711	0.000000	1.000000	1.000(
partner	4000.0	0.486750	0.499887	0.000000	0.000000	0.000(
promo_friends	4000.0	0.308500	0.461932	0.000000	0.000000	0.000(
phone	4000.0	0.903500	0.295313	0.000000	1.000000	1.000(

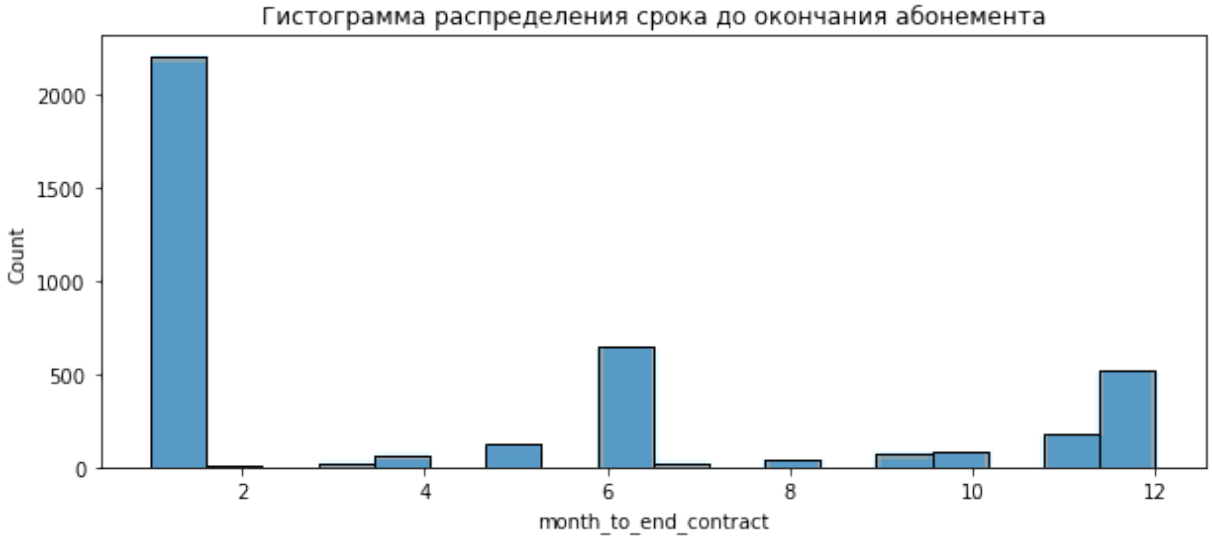
	count	mean	std	min	25%	5(
contract_period	4000.0	4.681250	4.549706	1.000000	1.000000	1.0000
group_visits	4000.0	0.412250	0.492301	0.000000	0.000000	0.0000
age	4000.0	29.184250	3.258367	18.000000	27.000000	29.0000
avg_additional_charges_total	4000.0	146.943728	96.355602	0.148205	68.868830	136.2201
month_to_end_contract	4000.0	4.322750	4.191297	1.000000	1.000000	1.0000
lifetime	4000.0	3.724750	3.749267	0.000000	1.000000	3.0000
avg_class_frequency_total	4000.0	1.879020	0.972245	0.000000	1.180875	1.8327
avg_class_frequency_current_month	4000.0	1.767052	1.052906	0.000000	0.963003	1.7195
churn	4000.0	0.265250	0.441521	0.000000	0.000000	0.0000

Вывод

- `gender` - посетители фитнес центра делятся примерно одинаково на мужчин и женьщин
- `near_location` - большинство посетителей проживают или работают в том же районе, что и фитнес центр
- `partner` - почти половина посетителей являются сотрудниками организаций-партнеров фитнес центра
- `promo_friends` - 30% посетителей пришли по промокоду "приведи друга"
- `phone` - 90% оставили свой номер телефона
- `contract_period` - переменнам принимает 3-и значения (1 - 6 - 12). Основная масса посетителей имеет месячный абонемент.
- `group_visits` - примерно 41% посещает групповые занятия
- `age` - возраст посетителей находится в интервале от 18 до 41 года. При этом средний возраст 29 лет и он практически совпадает с 50% перцентилем, что говорит нам о нормальном распределении скорее всего
- `avg_additional_charges_total` - средняя доп выручка 147 руб. При этом есть отличие между средним показанием и 50% перцентилем. Это говорит о смещении распределения.
- `month_to_end_contract` - эта переменная показывает количесвто месяцев до окончания действия абонемента. Ниже приведена гистограмма распределения. Можно наблюдать три пика.Самый большой пик это 1 месяц. Скорее всего это связанно с продажами абонементов различного срока действия или же такую неравномерность можно обьяснить переодичностью активной работы отдела продаж(маловероятно).
- `lifetime` - среднее время 3 месяца с момета первого обращения в клуб. Распределение ассиметричное и имеет смещение вправо. Гистограмма
- `avg_class_frequency_total` - в среднем клеб посещают 1,87 раз в неделю. Распределение ассиметричное и смещено вправо, хотя хвост не большой.
- `avg_class_frequency_current_month` - Среднее 1,76 посещений в последний месяц. Это говорит о том что скорее всего последний месяц посещаемость по какой-то причине упала
- `churn` - отток. 26% клуб "потерял" своих клиентов.

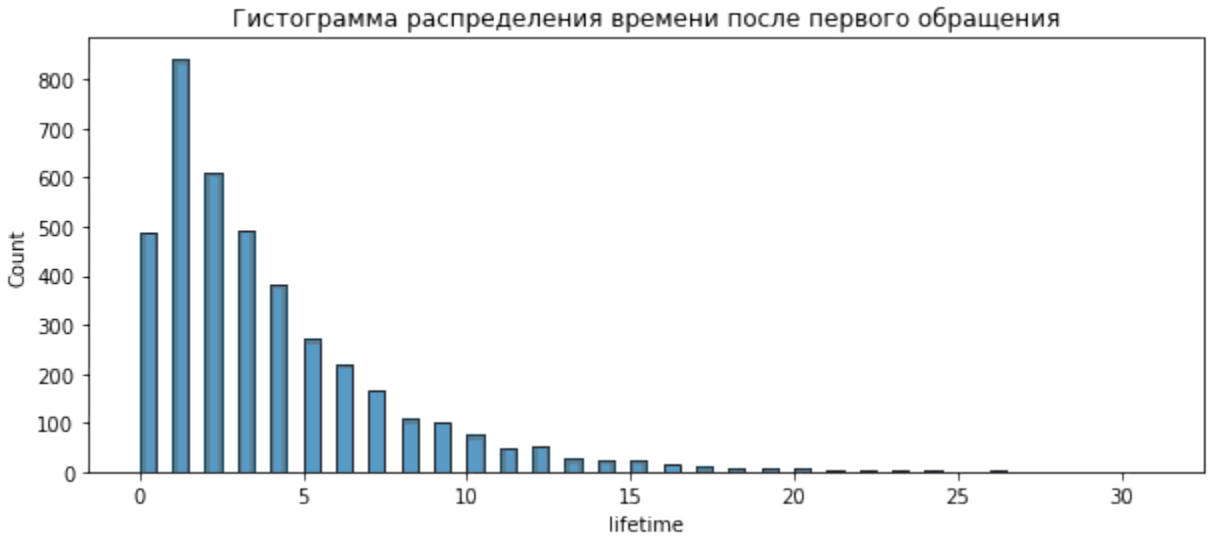
Гистограмма распределения срока до окончания абонемента

In [7]:
```
plt.figure(figsize=(10,4))
plt.title('Гистограмма распределения срока до окончания абонемента')
sns.histplot(df['month_to_end_contract'])
plt.show()
```



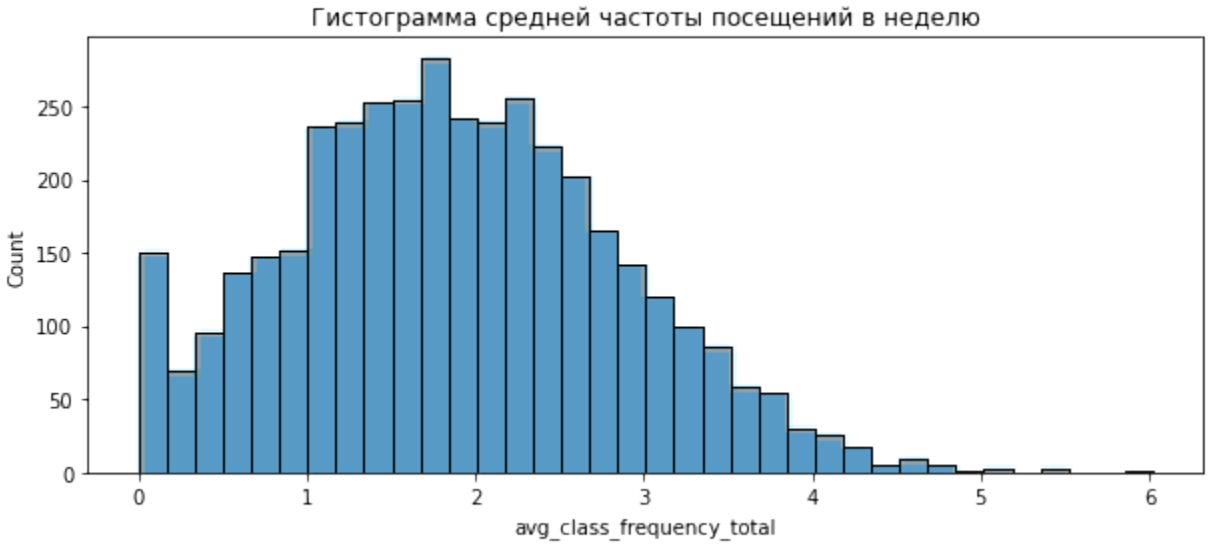
Гистограмма распределения времени после первого обращения

In [8]:
```
plt.figure(figsize=(10,4))
plt.title('Гистограмма распределения времени после первого обращения')
sns.histplot(df['lifetime'])
plt.show()
```



Гистограмма средней частоты посещений в неделю

In [9]:
```
plt.figure(figsize=(10,4))
plt.title('Гистограмма средней частоты посещений в неделю')
sns.histplot(df['avg_class_frequency_total'])
plt.show()
```



Посмотрите на средние значения признаков в двух группах — тех, кто ушел в отток и тех, кто остался

In [10]:
```
df.groupby(['churn']).mean().T
```

Out[10]:

churn	0	1
gender	0.510037	0.510839
near_location	0.873086	0.768143
partner	0.534195	0.355325
promo_friends	0.353522	0.183789
phone	0.903709	0.902922
contract_period	5.747193	1.728558
group_visits	0.464103	0.268615
age	29.976523	26.989632
avg_additional_charges_total	158.445715	115.082899
month_to_end_contract	5.283089	1.662582
lifetime	4.711807	0.990575
avg_class_frequency_total	2.024876	1.474995
avg_class_frequency_current_month	2.027882	1.044546

Вывод

- `gender` - средний возраст в группах не изменился
- `near_location` - среднее значение в группе "оттока" уменьшилось. Чем дальше от клуба тем вероятнее отток
- `partner` - среднее значение в группе "оттока" уменьшилось. Посетители-работники компаний партнеровполучают скидку и следовательно с меньшей вероятностью прекратят ходить в клуб.
- `promo_friends` - среднее значение в группе "оттока" уменьшилось. Посетители попавшие в клуб по рекомендации прекращают ходить в клую реже
- `phone` - значение не изменилось. Заполнение графы с телефоном не влияет на вероятность оттока

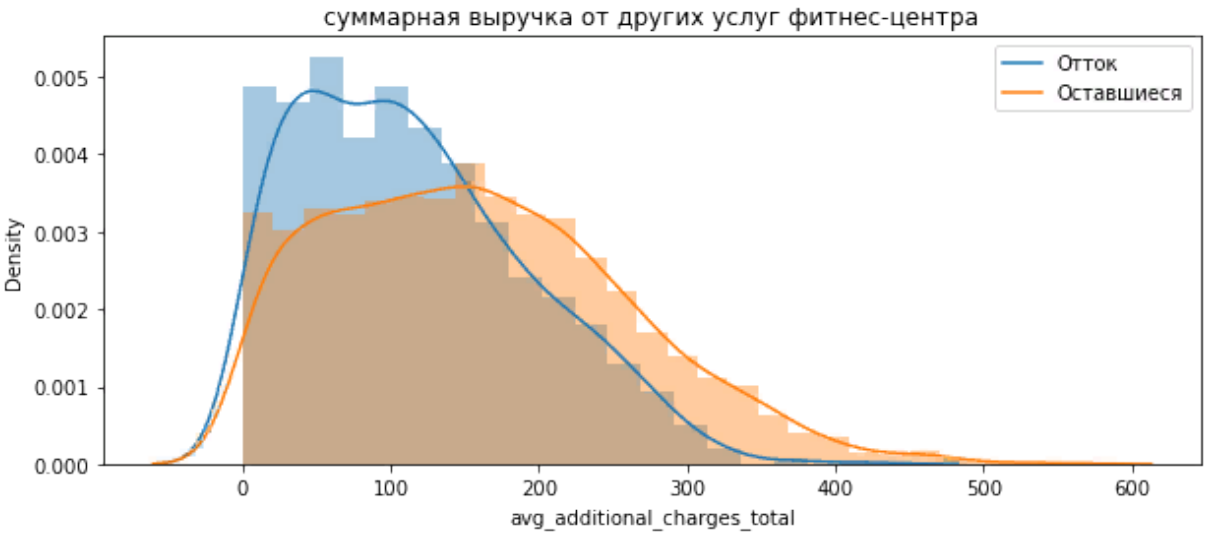
- `contract_period` - среднее значение в группе "оттока" уменьшилось. Но это мало о чем говорит, т.к. это категорезированная переменная.
- `group_visits` - среднее значение в группе "оттока" уменьшилось. Логично что клиенты которые посещают групповые занятия реже прекращают ходить в центр
- `age` - среднее значение в группе "оттока" уменьшилось. Более молодые чаще прекращают пользоваться клубом.
- `avg_additional_charges_total` - среднее значение в группе "оттока" уменьшилось. Те кто больше потребляет дополнительные услуги чаще продолжают ходить в клуб
- `month_to_end_contract` - среднее значение в группе "оттока" уменьшилось значительно. Скорее всего чаще прекращают ходить в клуб владельцы месячных абонементов.
- `lifetime` - среднее значение в группе "оттока" уменьшилось значительно. Скорее всего очень много тех, кто пришел просто попробовать.
- `avg_class_frequency_total` - среднее значение в группе "оттока" уменьшилось значительно. Кто реже посещает клуб, те чаще и прекращают им пользоваться
- `avg_class_frequency_current_month` - среднее значение в группе "оттока" уменьшилось значительно. Аналогичная ситуация с предыдущим пунктом.

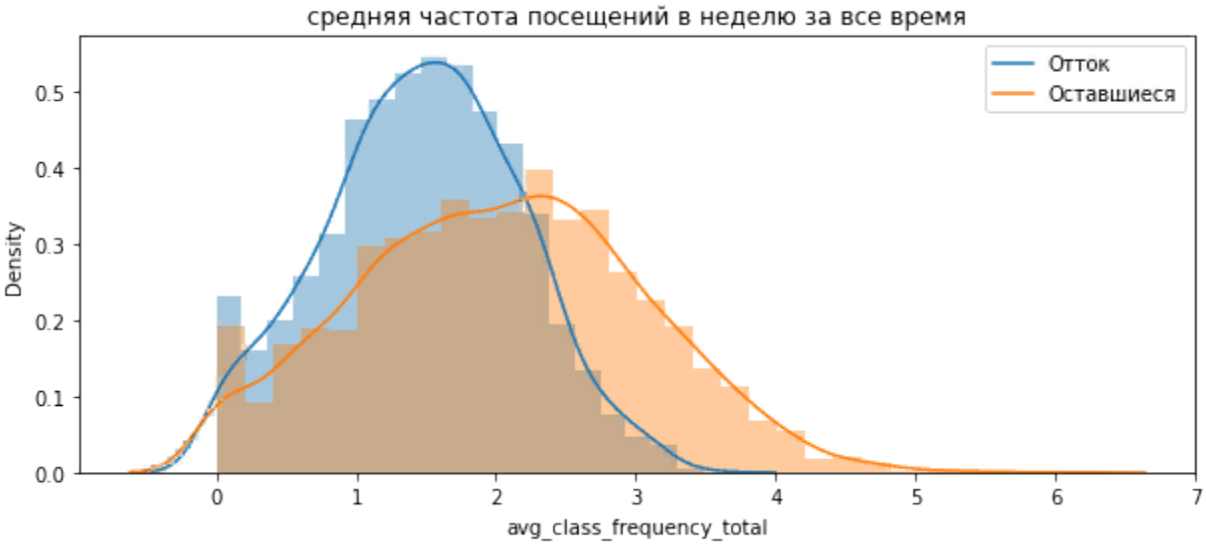
Постройте столбчатые гистограммы и распределения признаков для тех, кто ушёл (отток) и тех, кто остался (не попали в отток)

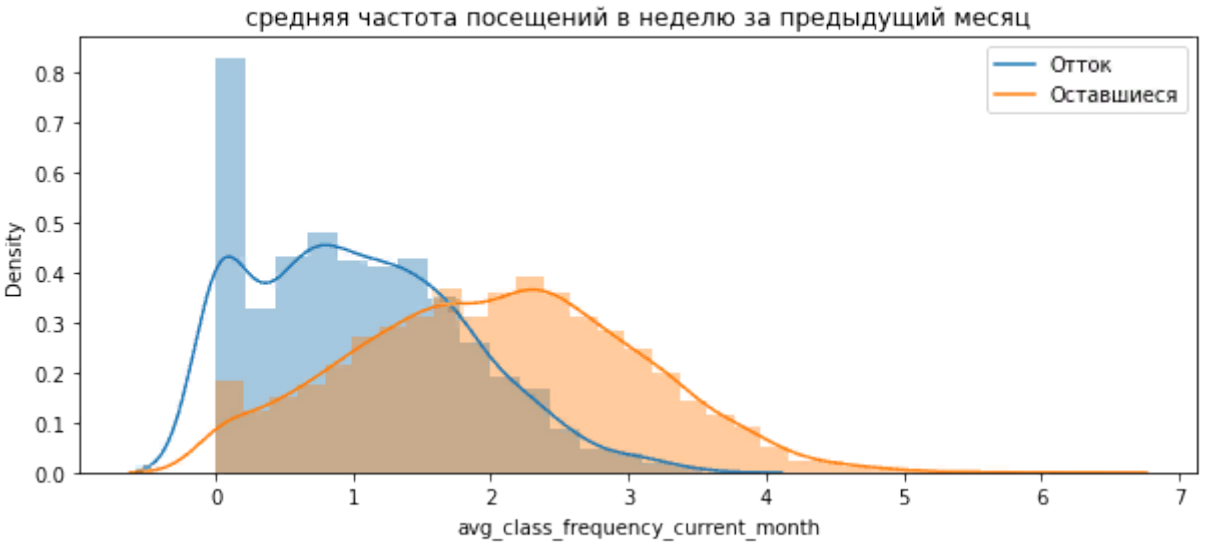
Все данные в файле gym_churn.csv можно разделить на количественные и категориальные. В первом случае будем использовать метод distplot() из библиотеки seaborn, в втором случае средние значения хорошо отобразит метод barplot из seaborn

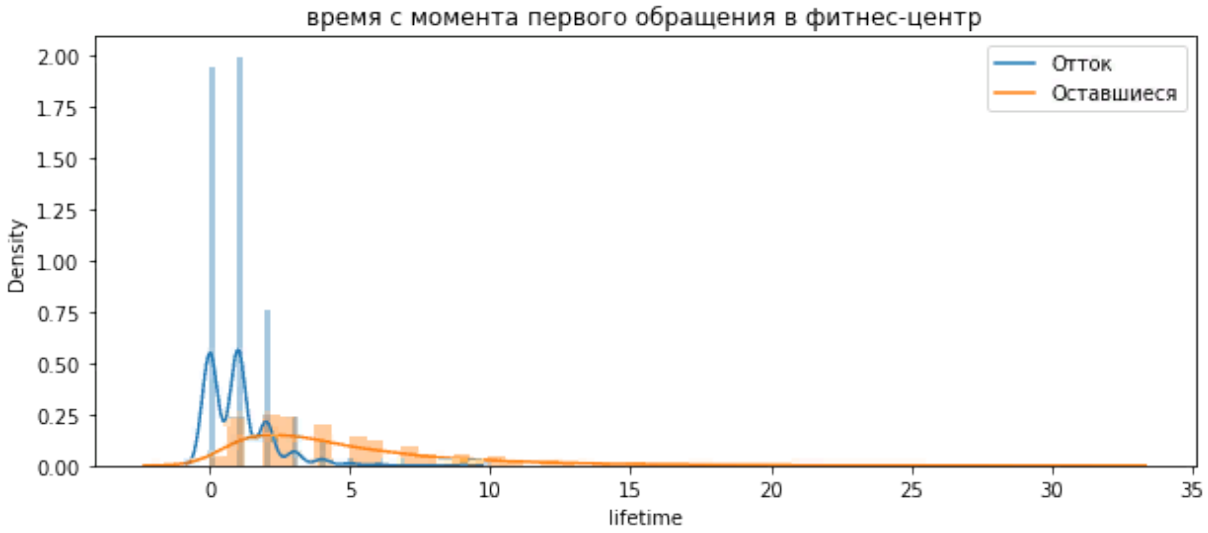
In [11]:

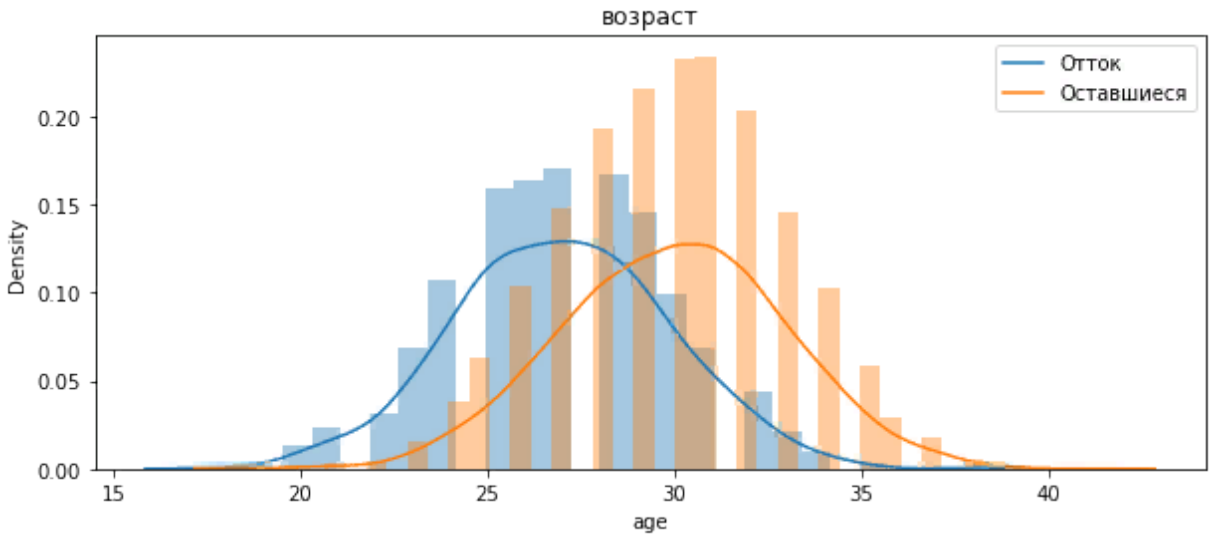
```
# для лучшей визуализации добавим словать перевода газвания столбцов
dict_columns={'gender':'пол',
              'near_location':'проживание или работа в районе, где находится
             'partner':'сотрудник компании-партнёра клуба',
             'promo_friends':'факт первоначальной записи в рамках акции «прив
             'phone':'телефон',
             'contract_period':'длительность текущего действующего абонемента
             'group_visits':'посещение групповых занятий',
             'age':'возраст',
             'avg_additional_charges_total':'суммарная выручка от других услу
             'month_to_end_contract':'количество месяцев до окончания абонеме
             'lifetime':'время с момента первого обращения в фитнес-центр',
             'avg_class_frequency_total':'средняя частота посещений в неделю
             'avg_class_frequency_current_month':'средняя частота посещений в
left = df[df['churn']==1]
stayed = df[df['churn']==0]
# список количественных переменных
distplot_columns = ['avg_additional_charges_total', 'avg_class_frequency_tota
                    'avg_class_frequency_current_month', 'lifetime', 'age',
                    'month_to_end_contract']
for column in distplot_columns:
    plt.figure(figsize=(10,4))
    plt.title(dict_columns[column])
    sns.distplot(left[column])
    sns.distplot(stayed[column])
    plt.legend(['Отток', 'Оставшиеся'])
    plt.show()
```

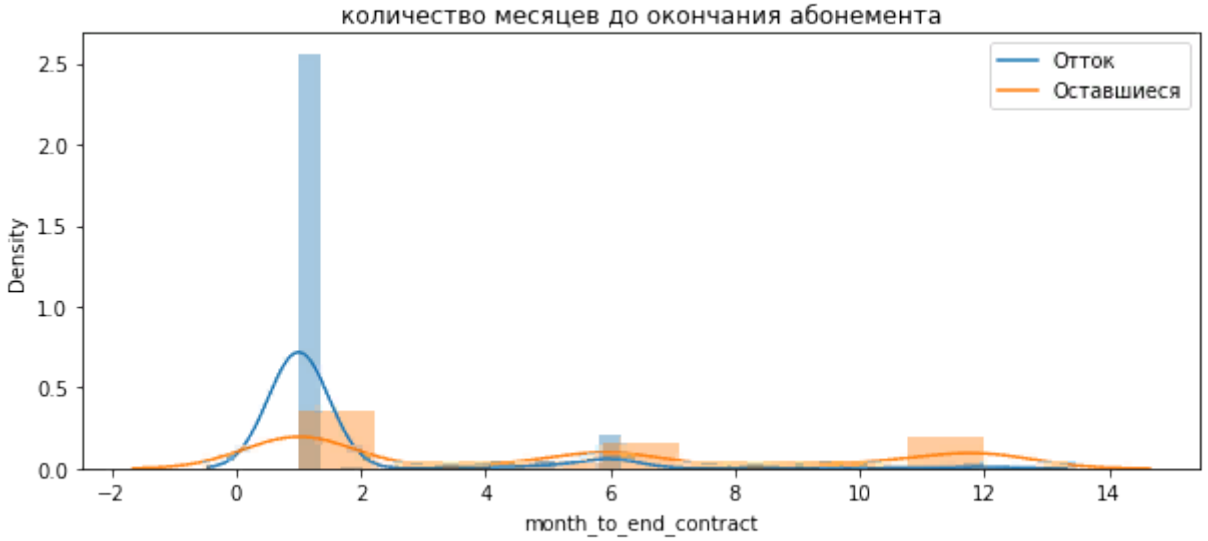

суммарная выручка от других услуг фитнес-центра
Отток
Оставшиеся
Density
0.005
0.004
0.003
0.002
0.001
0.000
0
100
200
300
400
500
600
avg_additional_charges_total


средняя частота посещений в неделю за все время
Отток
Оставшиеся
Density
0.5
0.4
0.3
0.2
0.1
0.0
0
1
2
3
4
5
6
7
avg_class_frequency_total


средняя частота посещений в неделю за предыдущий месяц
Отток
Оставшиеся
Density
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
1
2
3
4
5
6
7
avg_class_frequency_current_month

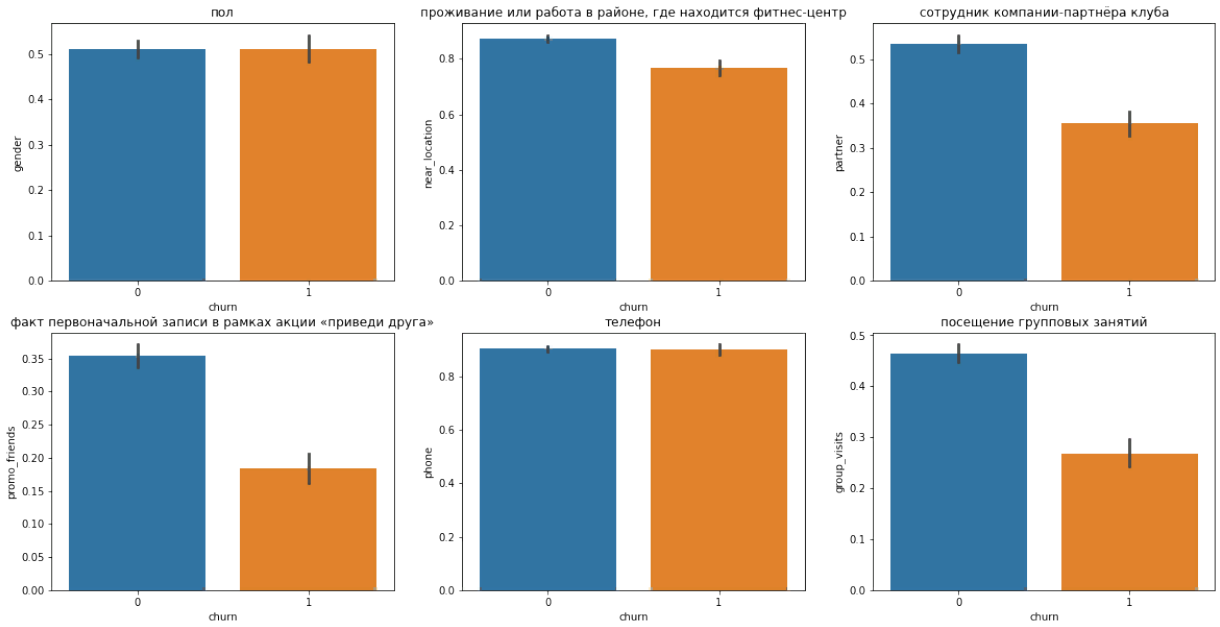






In [12]:

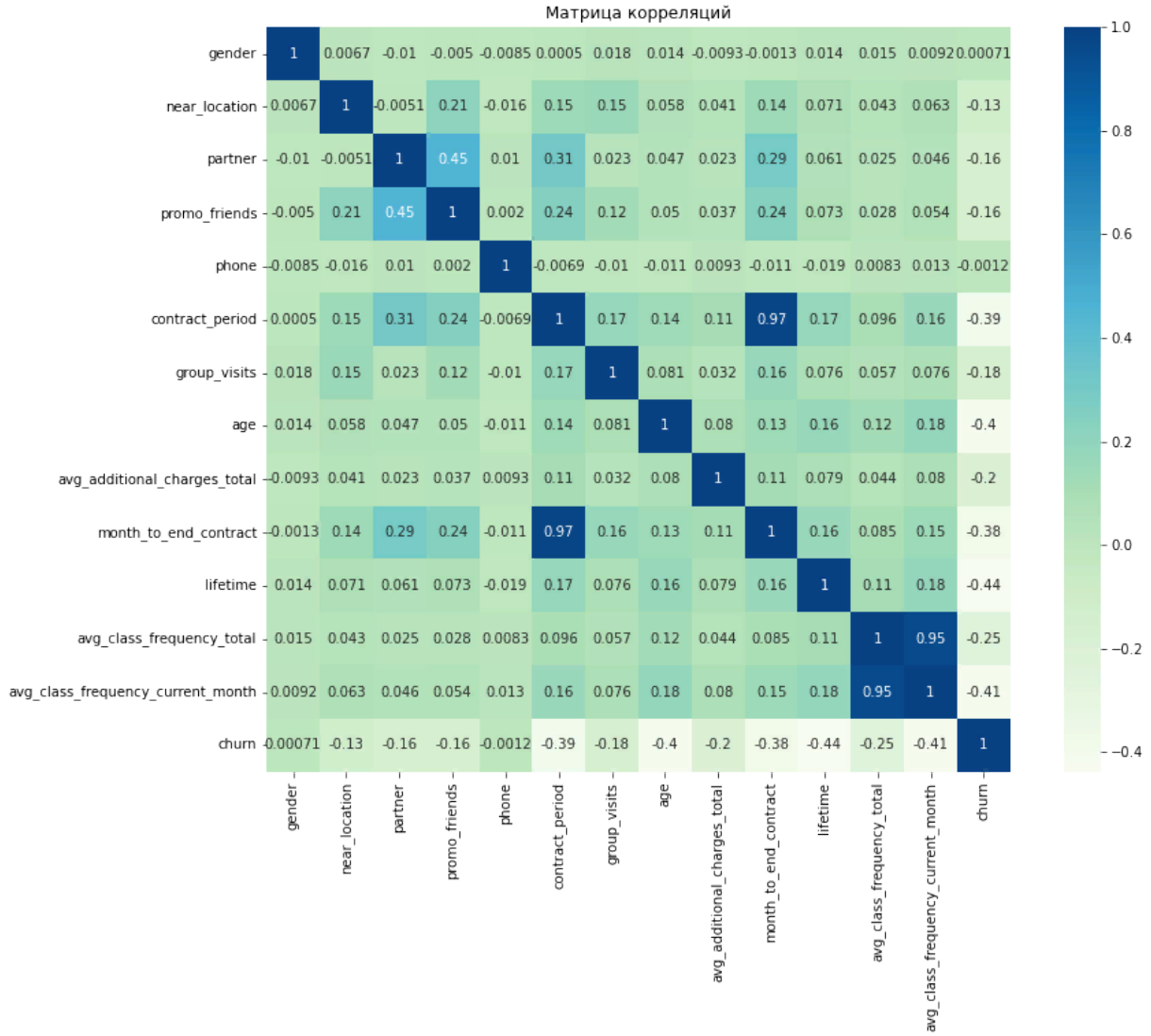
```
# графики качесственных переменных
n=1
plt.figure(figsize=(20, 10))
for column in dict_columns:
    if column in ['gender', 'near_location', 'partner', 'promo_friends', 'pho
        plt.subplot(2, 3, n)
        sns.barplot(x='churn', y=column, data=df)
        plt.title(dict_columns[column])
        n += 1
plt.show()
```



Постройте матрицу корреляций и отобразите её.

In [13]:

```
plt.figure(figsize=(14,10))
sns.heatmap(data = df.corr(), annot=True, square=True, cmap='GnBu')
plt.title('Матрица корреляций')
plt.show()
```



Вывод переменные между собой слабо корелируются. Единственная пара это `month_to_end_contract` - `contract_period`. Для линейных моделей взаимная корреляция нежелательна. Чтобы избавиться от мультиколлинеарности, удалим из датафрейма одну из переменных.

In [14]:
```
df.drop('contract_period', axis = 1, inplace = True)
```

Постройте модель прогнозирования оттока клиентов

Постройте модель бинарной классификации клиентов, где целевой признак — факт оттока клиента в следующем месяце:

Разбейте данные на обучающую и валидационную выборку функцией

Разбиваем выборку на обучающую и валидационную в пропорциях 80% к 20%

In [15]:
```
X = df.drop(['churn'], axis = 1)
y = df['churn']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, rand
```

Обучите модель на train-выборке двумя способами: логистической регрессией, случайным лесом.

In [16]:
```
# стандартизируем и нормализуем данные для обучения и валидации
scaler = StandardScaler()
X_train_st = scaler.fit_transform(X_train)
X_test_st = scaler.transform(X_test)

# обучим модель логической регрессии
lr_model = LogisticRegression(random_state=0)
lr_model.fit(X_train_st, y_train)

# обучим модель алгоритмам случайного леса
rf_model = RandomForestClassifier(n_estimators = 100, random_state = 0)
rf_model.fit(X_train_st, y_train)
```

Out[16]:
```
RandomForestClassifier(random_state=0)
```

Оцените метрики accuracy, precision и recall для обеих моделей на валидационной выборке. Сравните по ним модели. Какая модель показала себя лучше на основании метрик?

In [17]:
```
lr_predictions = lr_model.predict(X_test_st)
lr_probabilities = lr_model.predict_proba(X_test_st)[:,1]
print('Метрики для модели логистической регрессии:')
print('accuracy_score: {}\nprecision_score: {}\nrecall_score:{}'.format(
    accuracy_score(y_test, lr_predictions),
    precision_score(y_test, lr_predictions),
    recall_score(y_test, lr_predictions)))
```

```
Метрики для модели логистической регрессии:
accuracy_score: 0.925
precision_score: 0.8631578947368421
recall_score:0.8282828282828283
```

In [18]:

```
rf_predictions = rf_model.predict(X_test_st)
rf_probabilities = rf_model.predict_proba(X_test_st)[:,1]
print('\nМетрики для модели случайного леса:')
print('accuracy_score: {}\nprecision_score: {}\nrecall_score:{}'.format(
    accuracy_score(y_test, rf_predictions),
    precision_score(y_test, rf_predictions),
    recall_score(y_test, rf_predictions)))
```

```
Метрики для модели случайного леса:
accuracy_score: 0.9175
precision_score: 0.84375
recall_score:0.8181818181818182
```

Вывод Доля правильных прогнозов и полнота чуть выше в модели логистической регрессии. Таким образом, модель логистической регрессии показала себя лучше.

Выведем коэффициенты функции логической регрессии по степени важности

In [19]:

```
features = pd.DataFrame(lr_model.coef_.T, X.columns).reset_index()
features.columns = ['feature', 'coef']
features['coef'] = features['coef'].apply(lambda x: abs(x))
features = features.sort_values(by='coef', ascending=False)
print('\nКоэффициенты признаков в оптимальной функции логистической регрессии
print(features)
```

```
Коэффициенты признаков в оптимальной функции логистической регрессии:
                             feature      coef
11  avg_class_frequency_current_month  4.461517
9                            lifetime  3.846626
10         avg_class_frequency_total  3.301901
8              month_to_end_contract  1.233650
6                                 age  1.093233
7        avg_additional_charges_total  0.549632
5                         group_visits  0.394791
3                        promo_friends  0.271025
2                              partner  0.092234
1                        near_location  0.081896
0                               gender  0.012958
4                                phone  0.006273
```

Кластеризацию клиентов

Стандартизируйте данные

In [20]:

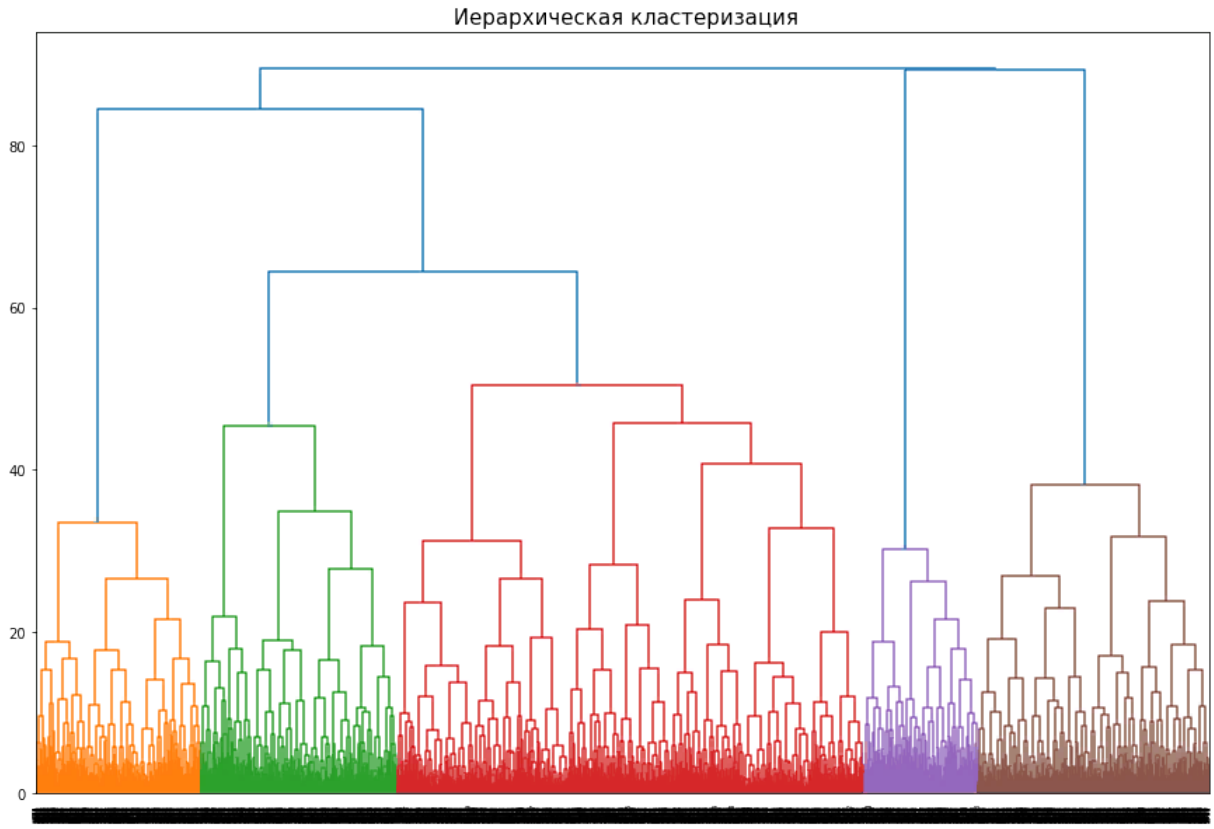
```
X_st = scaler.fit_transform(X)
```

Постройте матрицу расстояний функцией linkage() на стандартизованной матрице признаков и нарисуйте дендрограмму. На основании полученного графика предположите, какое количество кластеров можно выделить.

In [21]:

```
linked = linkage(X_st, method='ward')
plt.figure(figsize=(15, 10))
```

```
dendrogram(linked, orientation='top')
plt.title('\n Иерархическая кластеризация', fontsize=15)
plt.show()
```



Из графика `Иерархическая кластеризация` четко видны 5 кластеров

Обучите модель кластеризации на основании алгоритма K-Means и спрогнозируйте кластеры клиентов.

In [22]:
```
km = KMeans(n_clusters = 5, random_state = 0)
labels = km.fit_predict(X_st)
df['cluster'] = labels
```

Посмотрите на средние значения признаков для кластеров. Можно ли сразу что-то заметить?

In [23]:
```
df.groupby('cluster').mean().T
```

Out[23]:

cluster	0	1	2	3	4
gender	0.485597	0.524675	0.560428	0.493186	0.494382
near_location	0.992798	0.862338	0.973262	1.000000	0.000000
partner	0.940329	0.472727	0.309091	0.257240	0.486891
promo_friends	0.912551	0.306494	0.083422	0.094549	0.074906
phone	1.000000	0.000000	0.998930	1.000000	1.000000
group_visits	0.536008	0.425974	0.485561	0.330494	0.228464
age	29.612140	29.283117	30.270588	28.210392	28.573034
avg_additional_charges_total	154.221687	144.240418	165.107405	132.180078	136.299693

cluster	0	1	2	3	4
month_to_end_contract	7.080247	4.457143	4.655615	2.480409	2.674157
lifetime	4.432099	3.922078	4.989305	2.437819	2.910112
avg_class_frequency_total	1.868932	1.846575	2.837277	1.226093	1.678385
avg_class_frequency_current_month	1.827433	1.716160	2.836336	1.001368	1.504945
churn	0.103909	0.267532	0.048128	0.500852	0.419476

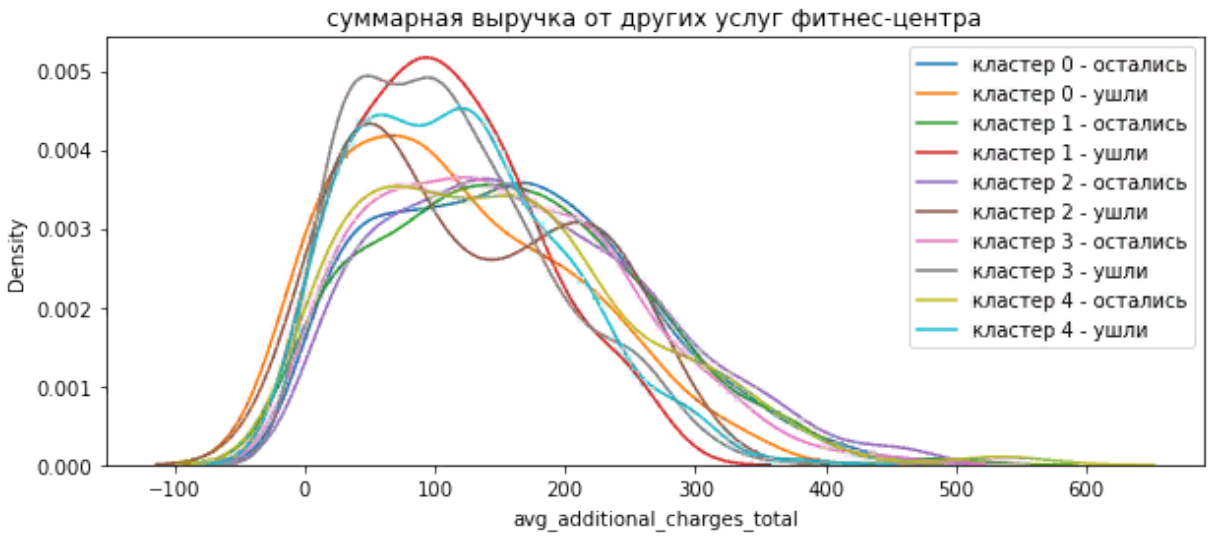
Вывод Если обратить внимание на строчку `churn`, то в среднем самый большой отток клиентов наблюдается в группе номер 3, почти 50%. По этой группе можно отметить следующее:

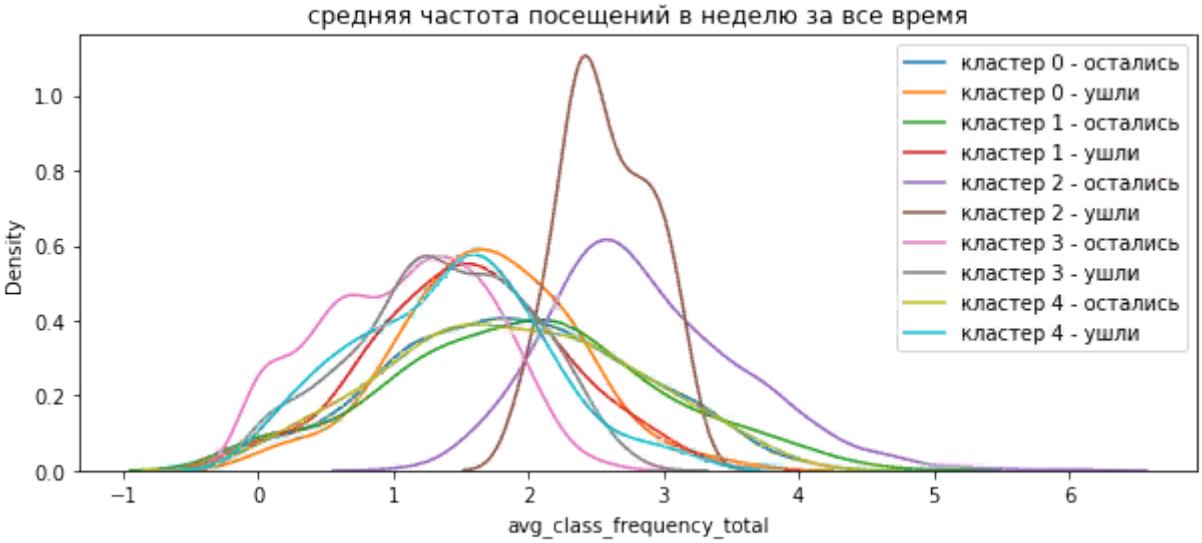
- самый низкий частота посещения в месяц и в общем
- меньше всего проведенное время
- меньше всего потрачено на доп.услуги

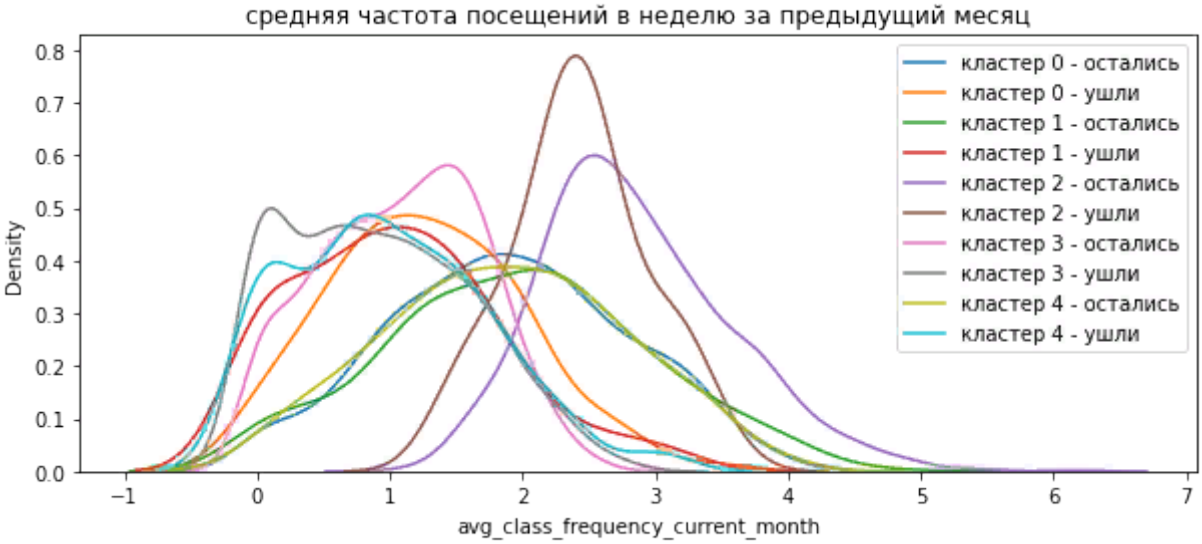
Постройте распределения признаков для кластеров. Можно ли что-то заметить по ним?

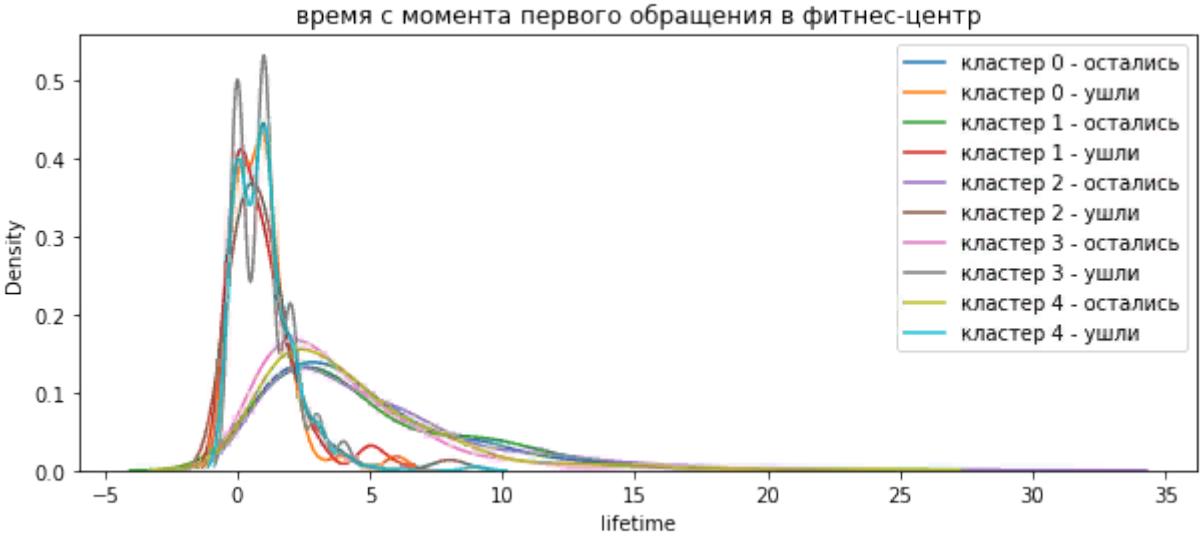
In [24]:
```
dict_churn = {1:'ушли', 0:'остались'}
def plot_claster_churn(cols):
    plt.figure(figsize=(10,4))
    for x in range(5):
        for ch in df['churn'].unique():
            sns.distplot(df.query('cluster==@x and churn == @ch')[cols],
                         hist=False, label = 'кластер {} - {}'.format(x, dict_chu
    plt.legend()
    plt.title(dict_columns[cols])
    plt.show()
```

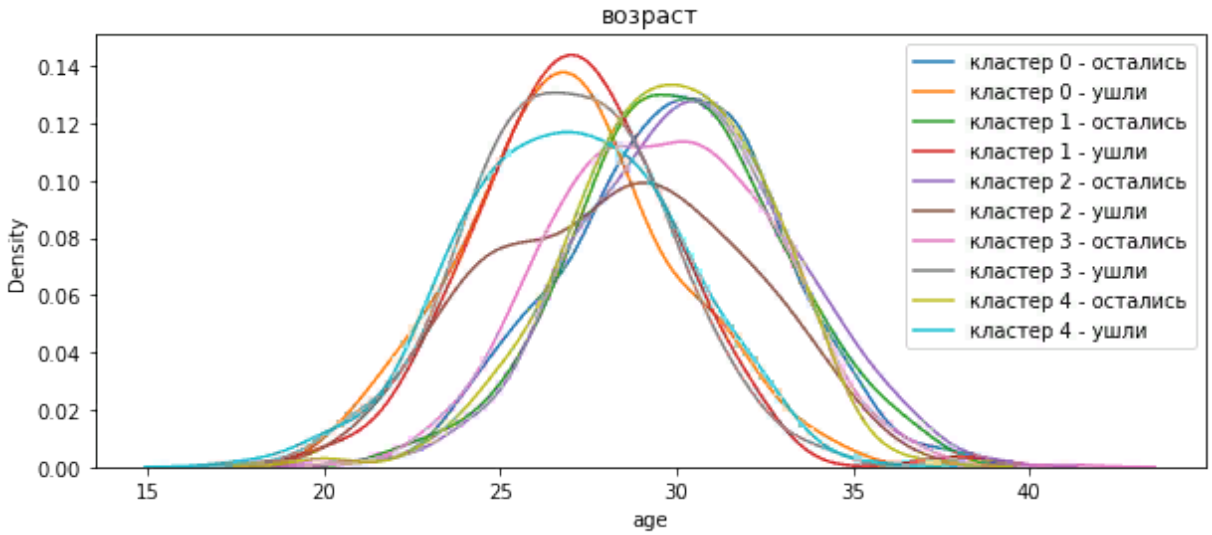
In [25]:
```
for _ in distplot_columns:
    plot_claster_churn(_)
```

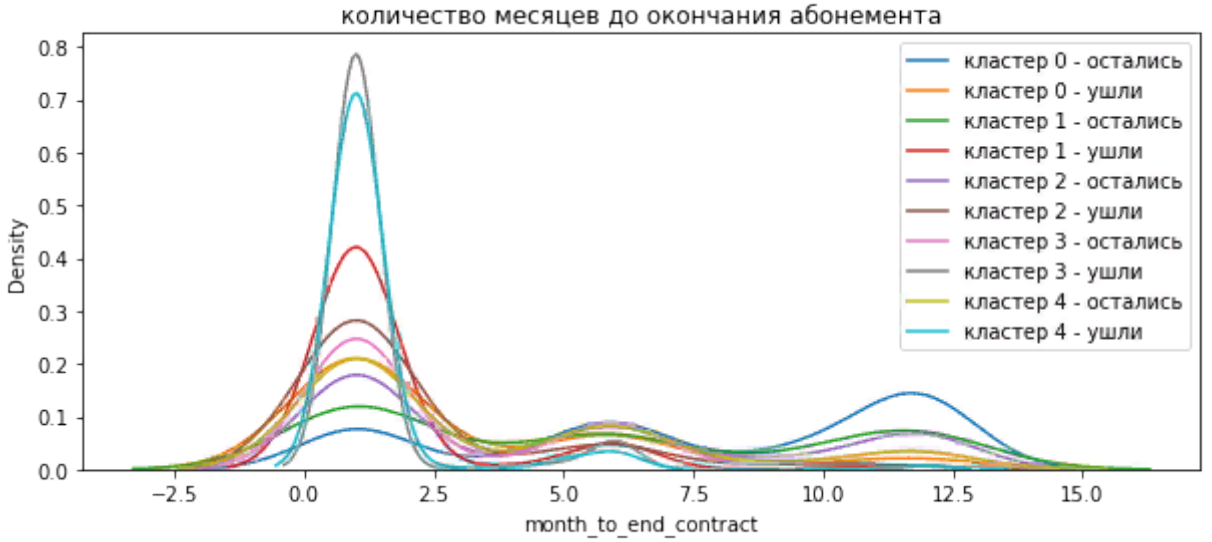



средняя частота посещений в неделю за все время
кластер 0 - остались
кластер 0 - ушли
кластер 1 - остались
кластер 1 - ушли
кластер 2 - остались
кластер 2 - ушли
кластер 3 - остались
кластер 3 - ушли
кластер 4 - остались
кластер 4 - ушли
Density
avg_class_frequency_total


средняя частота посещений в неделю за предыдущий месяц
кластер 0 - остались
кластер 0 - ушли
кластер 1 - остались
кластер 1 - ушли
кластер 2 - остались
кластер 2 - ушли
кластер 3 - остались
кластер 3 - ушли
кластер 4 - остались
кластер 4 - ушли
Density
avg_class_frequency_current_month


время с момента первого обращения в фитнес-центр
кластер 0 - остались
кластер 0 - ушли
кластер 1 - остались
кластер 1 - ушли
кластер 2 - остались
кластер 2 - ушли
кластер 3 - остались
кластер 3 - ушли
кластер 4 - остались
кластер 4 - ушли
Density
lifetime

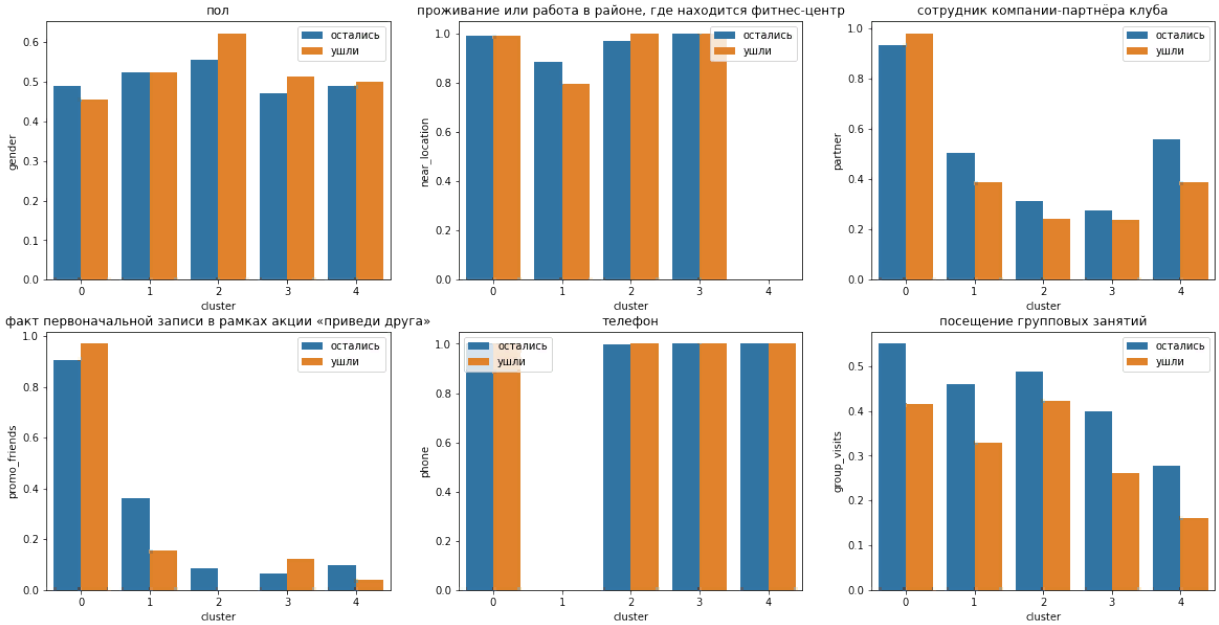




In [26]:

```
category_varible_table = df.groupby(['cluster','churn']).mean().reset_index()
category_varible_table['churn'].replace(0, 'остались', inplace=True)
category_varible_table['churn'].replace(1, 'ушли', inplace=True)
plt.figure(figsize=(20, 10))
n=1
for column in ['gender', 'near_location', 'partner', 'promo_friends', 'phone'
    plt.subplot(2, 3, n)
    sns.barplot(x='cluster',
                    y=column,
                    hue='churn',
                    data = category_varible_table,
                   )
    plt.legend()
    plt.title(dict_columns[column])
    n += 1


plt.show()
```



Вывод В каждом кластере сложно выделить фактор который бы сильно оказывал влияние на отток клиентов. По всем кластерам можно набдюдать приблизительно одинаковое поведение графиков. Если растет показатель оттока, одновременно растет и показатель тех кто остался

Для каждого полученного кластера посчитайте долю оттока (методом groupby()). Отличаются ли они по доле оттока? Какие кластеры склонны к оттоку, а какие — надёжны?

In [27]:
```
df.groupby('cluster')['churn'].mean().sort_values(ascending = False)
```

Out[27]:
```
cluster
3    0.500852
4    0.419476
1    0.267532
0    0.103909
2    0.048128
Name: churn, dtype: float64
```

Вывод Как видно из таблицы выше лидеры по доли оттоку являются 3 и 4 кластер

Общий вывод

В процессе иследования были проанализированы данные посещения фитнес-центра. Мы построили пронозную модель с неплохой точностью. Теперь мы можем прогнозировать вероятность оттока по каждому клиенту. Явных зависимостей факта оттока от одного конкретного признака из рассматриваемых не обнаружено. Наибольшее влияние оказывают `avg_class_frequency_current_month`, `lifetime`, `avg_class_frequency_total`. При этом по ним можно оценить только клиентов которые имеют уже какую либо историю посещений. Но интересно было бы ответить на вопрос о потенциальном уходе клиента в момент обращения в клуб. В этом нам может помочь разбивка клиентов на кластеры / группы. Можно выжедить 5 групп Кластер 0 (близ живущие)

- Отток - 10%

- Живут или работают недалеко от фитнес-центра
- В основном обладатели долгосрочных абонементов
- Посещают клуб достаточно часто 1-2 раза в неделю
- Пользуются клубом достаточно давно
- Скорее всего являются сотрудниками организаций партнеров или воспользовались промоакцией

Кластер 1 (сотрудники компаний партеров + акции)

- Отток - 26%
- Живут или работают недалеко от фитнес-центра
- Посещают клуб достаточно часто 1-2 раза в неделю
- В основном обладатели долгосрочных абонементов
- Посещают клуб достаточно часто 1-2 раза в неделю
- Пользуются клубом достаточно давно
- Скорее всего являются сотрудниками организаций партнеров

Кластер 2 (постоянные клиенты)

- Отток - 48%
- Живут или работают недалеко от фитнес-центра
- Купили абонемент за полную стоимость
- Больше всего тратят деньги на доп услуги
- Часто посещают клуб 2-3 раза в неделю
- Пользуются клубом достаточно давно

Кластер 3 (студенты)

- Отток - 50%
- Самые молодые
- Проводят в клубе мало времени 1 раз в неделю. Самый низкий показатель
- Пользуются не долгострочными абонементами
- Мало тратят денег на доп услуги
- Редко пользуются групповыми занятиями
- Не являются сотрудниками компаний партнеров
- Пришли не по акции

Класте 4 (далеко живут)

- Отток - 41%
- Живут далеко
- Проводят в клубе мало времени 1 раз в неделю.
- Пользуются не долгострочными абонементами
- Мало тратят денег на доп услуги
- Редко пользуются групповыми занятиями
- Не являются сотрудниками компаний партнеров

- Пришли не по акции

Рекомендации: для простоы воспринимания кластеров сотрудниками фитнесс клуба, каждому кластеру присвоено название условно характеризующее модель поведениия посетителя. Больше всего теряют группы "студент" и "далеко живут". Отделу маркетига можно рассмотреть различные программы для увеличения привлекательности клуба для этих категорий.
Так же необходим мониторинг за такими показателями как `avg_class_frequency_current_month` и `avg_class_frequency_total`. Падение этих показателей может говорить о потери интереса к клубу и возможному уходу клиента.

In [ ]: