Стратегия взаимодействия с клиентами фитнесс - центров "Культурист - Датасаентист"

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. 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.

Шаг. Загрузка и описание данных.

«Культурист-датасаентист» предоставил сведения в csv-файлах. Заказчик подготовил данные, которые содержат данные на месяц до оттока и факт оттока на определённый месяц. Набор данных включает следующие поля:

- 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, 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]:
```
# подключение необходимых библиотек

import pandas as pd
import numpy as np

from matplotlib import pyplot as plt
%matplotlib inline
import seaborn as sns
from plotly import graph_objects as go


from scipy.cluster.hierarchy import dendrogram, linkage

from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, roc_a

from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.cluster import KMeans
```

In [2]:
```
import warnings
warnings.filterwarnings('ignore')
```

In [3]:
```
# читаем данные из файла

try:
    df = pd.read_csv('gym_churn.csv')
except:
    df = pd.read_csv('/datasets/gym_churn.csv')
df.head()
```

Out[3]:

	gender	Near_Location	Partner	Promo_friends	Phone	Contract_period	Group_visits	Age	Avg_additional_charg
0	1	1	1	1	0	6	1	29	14
1	0	1	0	0	1	12	1	31	113
2	0	1	1	0	1	1	0	28	129
3	0	1	1	1	1	12	1	33	62
4	1	1	1	1	1	1	0	26	198

In [4]:
```
# приведем названия столбцов к нижнему регистру
df.columns = df.columns.str.lower()
```

Шаг. Исследовательский анализ данных (EDA)

In [5]:
```
# просмотр общей информации о DataFram-е
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
```

In [6]:
```
df.duplicated().sum()
```

Out[6]:
```
0
```

In [7]:
```
df['month_to_end_contract'].unique()
```

Out[7]:
```
array([ 5., 12.,  1.,  6.,  3., 10., 11.,  9.,  4.,  7.,  8.,  2.])
```

В предоставленных для анализа данных имеются 4000 записей о клиентах. Пропуски отсутствуют. Явных дубликатов не выявлено.

Столбцы `gender`, `near_location`, `partner`, `promo_friends`, `phone`, `group_visits`, `churn` могут содержать только значения 0 / 1, по смыслу целесообразно привести у типу данных `boolean`.

Столбец `month_to_end_contract` целесообразно привести к `int`, т.к. фактически в нем содержатся только целые числа.

In [8]:
```
# замена типа данных в столбцах DataFrame
df['month_to_end_contract'] = df['month_to_end_contract'].astype('int')
df[['gender', 'near_location', 'partner', 'promo_friends', 'phone', 'group_visits', 'churn
df[['gender', 'near_location', 'partner', 'promo_friends', 'phone', 'group_visits', 'churn
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   bool   
 1   near_location                      4000 non-null   bool   
 2   partner                            4000 non-null   bool   
 3   promo_friends                      4000 non-null   bool   
 4   phone                              4000 non-null   bool   
 5   contract_period                    4000 non-null   int64
```

```
 6   group_visits                       4000 non-null   bool
 7   age                                4000 non-null   int64
 8   avg_additional_charges_total       4000 non-null   float64
 9   month_to_end_contract              4000 non-null   int64
 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   bool
dtypes: bool(7), float64(3), int64(4)
memory usage: 246.2 KB
```

In [9]:
```
df.describe()
```

Out[9]:

	contract_period	age	avg_additional_charges_total	month_to_end_contract	lifetime	avg_class_freq
count	4000.000000	4000.000000	4000.000000	4000.000000	4000.000000	
mean	4.681250	29.184250	146.943728	4.322750	3.724750	
std	4.549706	3.258367	96.355602	4.191297	3.749267	
min	1.000000	18.000000	0.148205	1.000000	0.000000	
25%	1.000000	27.000000	68.868830	1.000000	1.000000	
50%	1.000000	29.000000	136.220159	1.000000	3.000000	
75%	6.000000	31.000000	210.949625	6.000000	5.000000	
max	12.000000	41.000000	552.590740	12.000000	31.000000	

Посмотрев общую информацию о таблице можем сказать следующее:

- `gender` в выборке приблизительно поровну мужчин и женщин;
- `near_location` большинство клиентов (порядка 85%) работают или проживают в районе фитнес центра, об этом свидетельствуют значения среднего на уровене 0,84 и первого квартиля 1;
- `partner` сотрудников компаний - партнеров и остальных клиентов примерно поровну;
- `promo_friends` по акции "приведи друга" занимаются порядка 30% клиентов;
- `phone` контактный телефон предоставили порядка 90% клиентов;
- `contract_period` средняя длительность действующего абонемента составляет 4-5 месяцев в 25% случаев клиенты покупали более длительные абонементы (от 6 до 12 месяцев);
- `group_visits` групповыми тренировками пользуются менее половины клиентов, можно предположить, что порядка 25 - 40%;
- `age` наш фитнес центр посещают клиенты в возрасте от 18 лет до 41 года, средний возраст нашего клиента 29 лет;
- `avg_additional_charges_total` средняя суммарная выручка от дополнительных услуг составляет 136 - 146 денег (единица измерения нам неизвестна);
- `month_to_end_contract` в среднем у клиентов остаток срока абонемента наблюдаем на уровне 4 месяцев, однако у больше половины клиентов остался всего 1 оплаченный месяц занятий;

- `lifetime` у нас имеются данные о клиентах, которые обращались в фитнес центр впервые от 0 до 31 месяца назад, среднее значение здесь наблюдаем на уровне 3-4 месяцев;
- `avg_class_frequency_total` можно сказать, что в среднем за неделю клиенты посещают клуб дважды, однако есть и большие любители фитнеса, занимающиеся 6 раз в неделю;
- `avg_class_frequency_current_month` данные о среднем количестве посещений на горизонте 1 месяца мало отличаются от данных за весь период;
- `churn` тех кто продолжает пользоваться услугами фитнес центра очевидно больше в среднем ушли менее 25% клиентов (1 - клиент ушел, 0 - продолжает пользоваться).

Далее посмотрим средние значения в разбивке по оставшимся и ушедшим клиентам.

In [10]:
```
round (df.groupby('churn').mean(), 2)
```

Out[10]:

churn	gender	near_location	partner	promo_friends	phone	contract_period	group_visits	age	avg_additional_ch
False	0.51	0.87	0.53	0.35	0.9	5.75	0.46	29.98	
True	0.51	0.77	0.36	0.18	0.9	1.73	0.27	26.99	

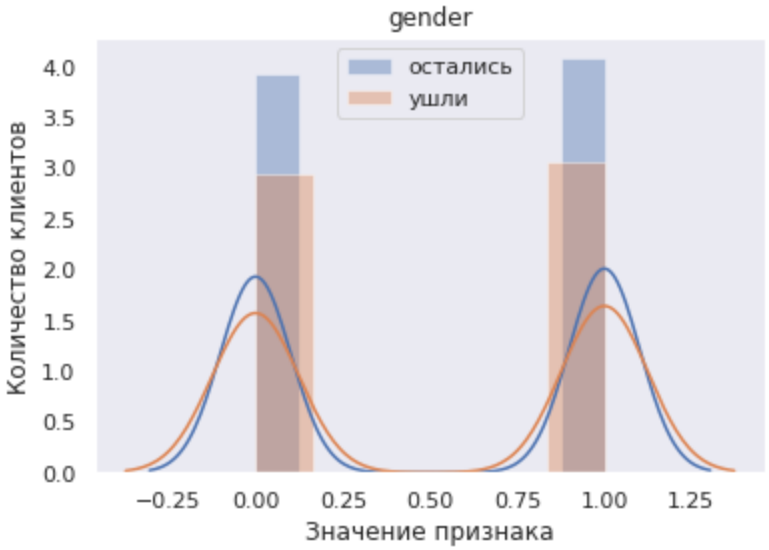
Выделим наиболее интересные наблюдения по средним значениям в полученных выборках:

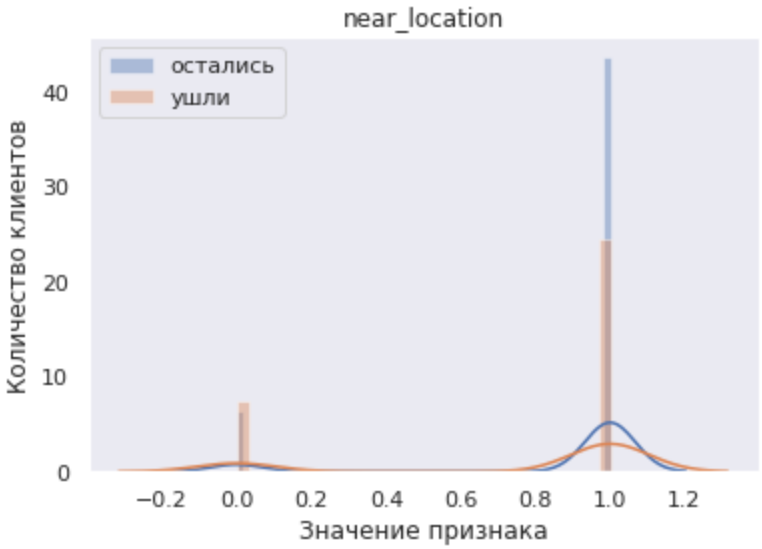
- клиенты проживающие в районе фитнес центра чуть лояльнее (среди оставшихся 87% местных, среди ушедших 77% местных);
- клиенты партнеров также несколько лояльнее (среди оставшихся 53% сотрудники партнеров, среди ушедших 36%);
- аналогичное наблюдение по акции приведи друга (35% оставшихся пришли по рекомендации, среди ушедших 18% пользовались этой акцией)
- ушедшие клиенты в среднем приобретали абонементы на меньший срок 1,73 против 5,75 у оставшихся;
- действующие клиенты в среднем чаще посещали групповые занятия 46% против 27% у ушедших;
- наблюдается небольшая разница в среднем возрасте около 30 лет у оставшихся клиентов и 27 лет у ушедших;
- доп услугами пользовались обе группы клиентов, но потратили немного больше действующие клиенты;
- и наконец действующие клиенты ожидаемо совершают большее количество еженедельных визитов (около 2) среднее же потерявшихся клиентов падает с 1.5 на всем периоде до 1 раза в неделю за последний месяц.

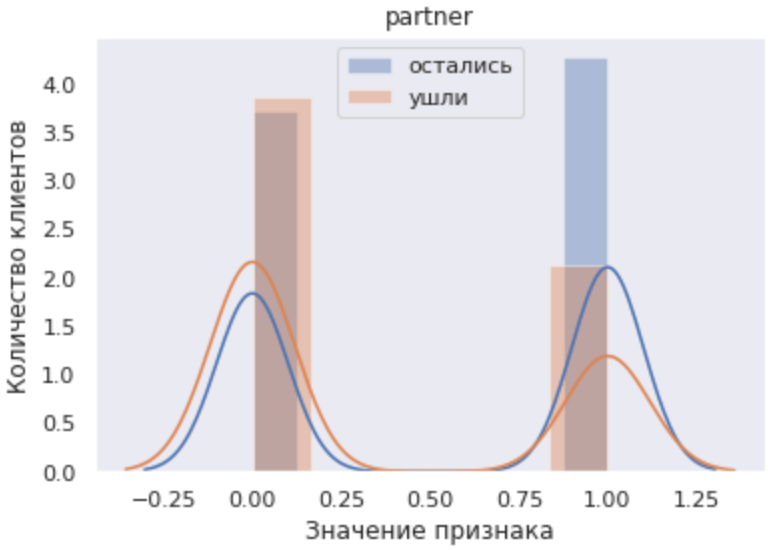
Далее построим гистограммы и распределения признаков для тех, кто ушел (отток) и тех, кто остался

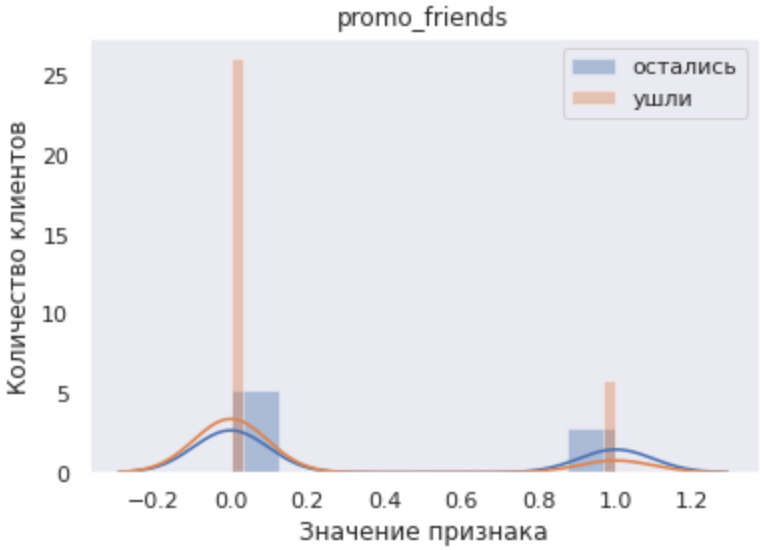
In [11]:
```
sns.set()
fig, ax_lst = plt.subplots(1, 1)


for i in df.drop('churn', axis=1).columns:

    sns.distplot(df.loc[df['churn'] == 0, i], label='остались')

    sns.distplot(df.loc[df['churn'] == 1, i], label='ушли')
```

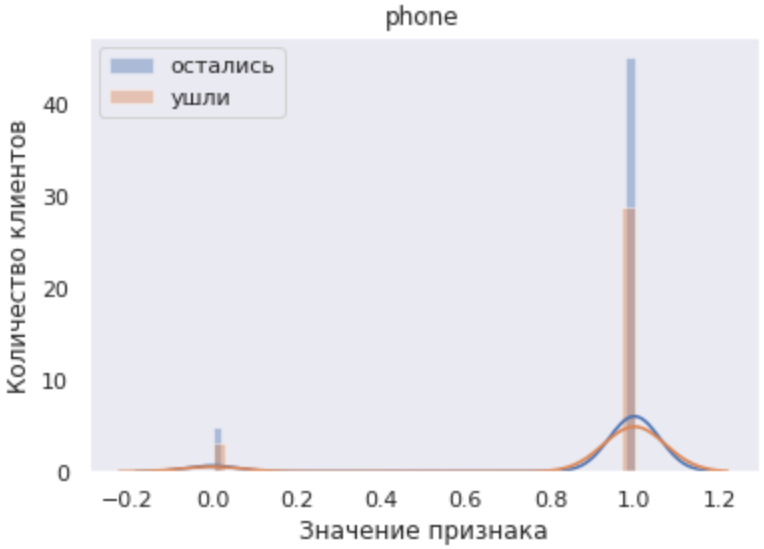
```
    plt.ylabel('Количество клиентов')
    plt.xlabel('Значение признака')
    plt.grid()
    plt.title(f'{i}')
    plt.legend()
    plt.show()
```

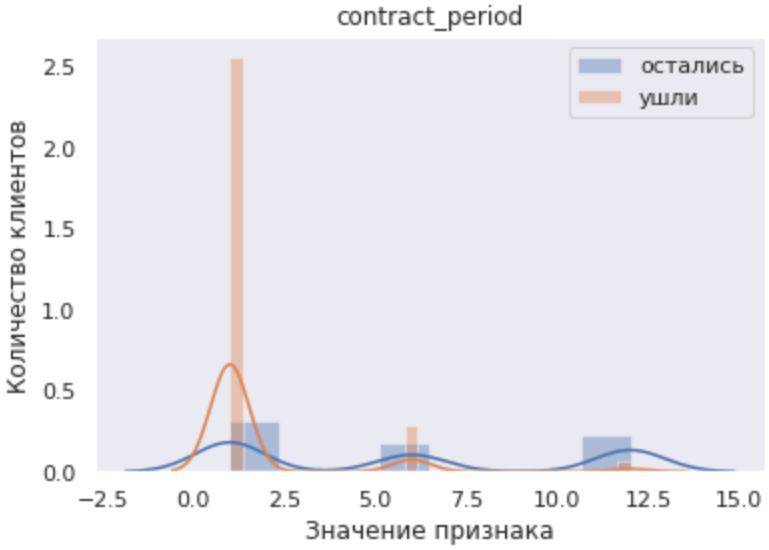


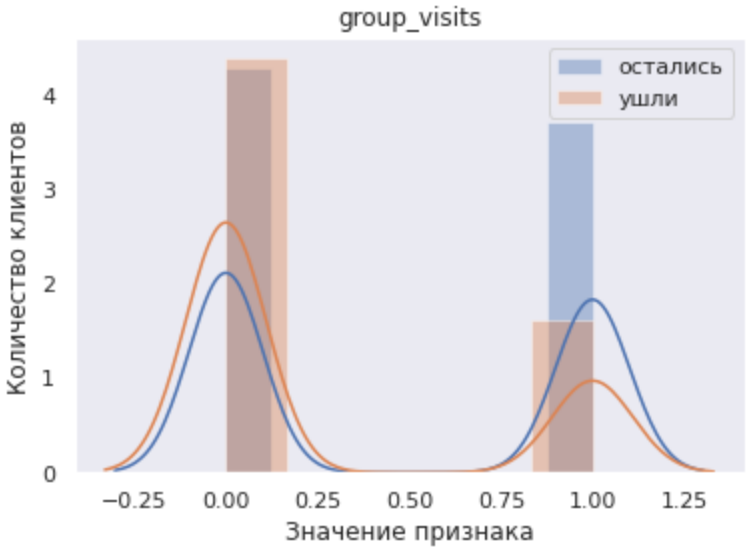


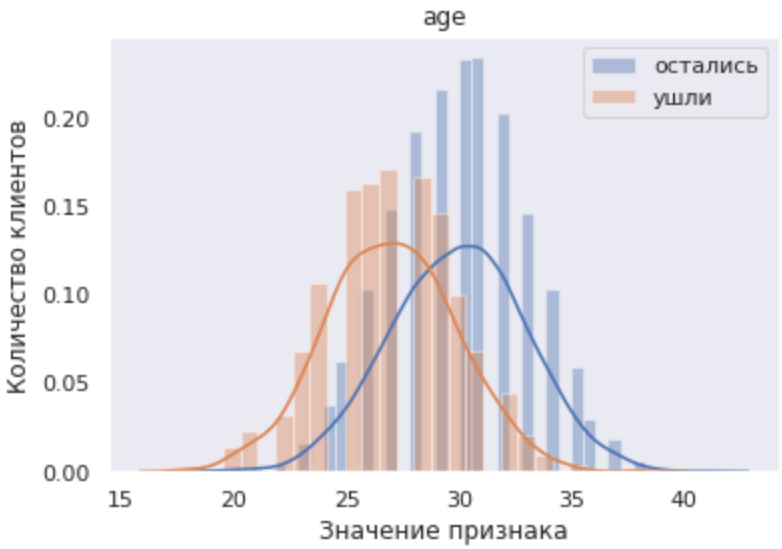


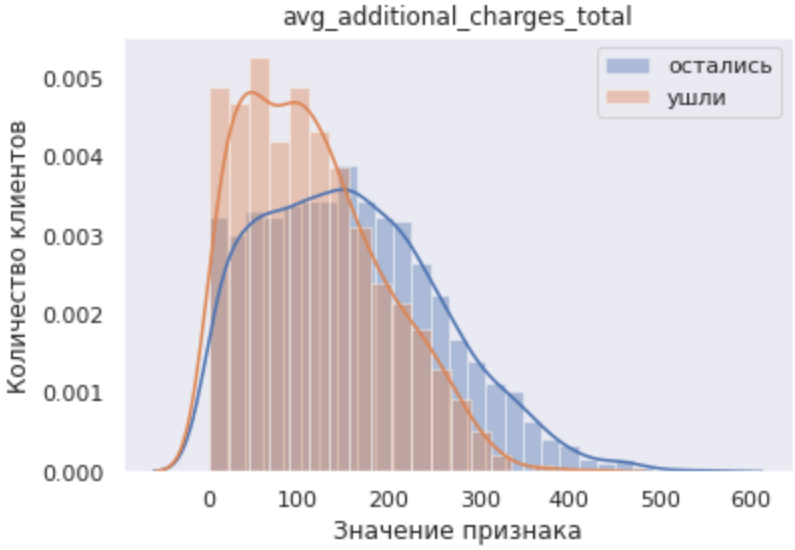

promo_friends
остались
ушли
Количество клиентов
Значение признака


phone
остались
ушли
Количество клиентов
Значение признака

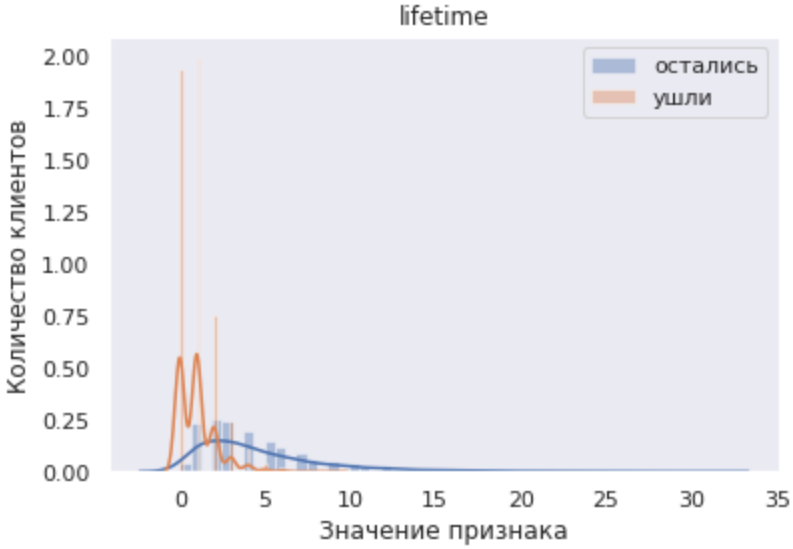

contract_period
остались
ушли
Количество клиентов
Значение признака

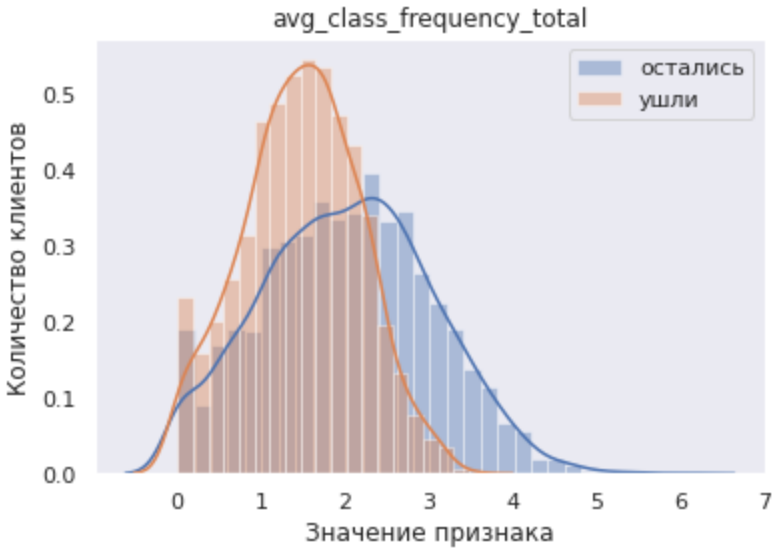

group_visits
остались
ушли
Количество клиентов
Значение признака

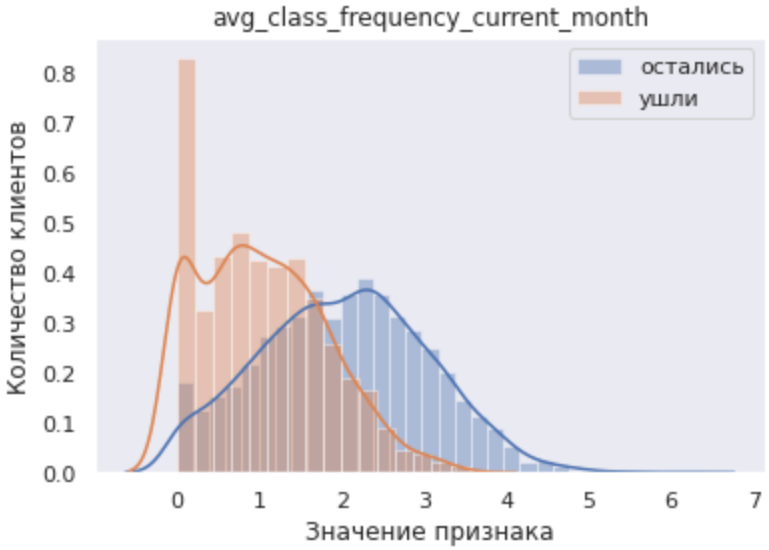

age
остались
ушли
Количество клиентов
Значение признака


avg_additional_charges_total
остались
ушли
Количество клиентов
Значение признака


month_to_end_contract
остались
ушли
Количество клиентов
Значение признака

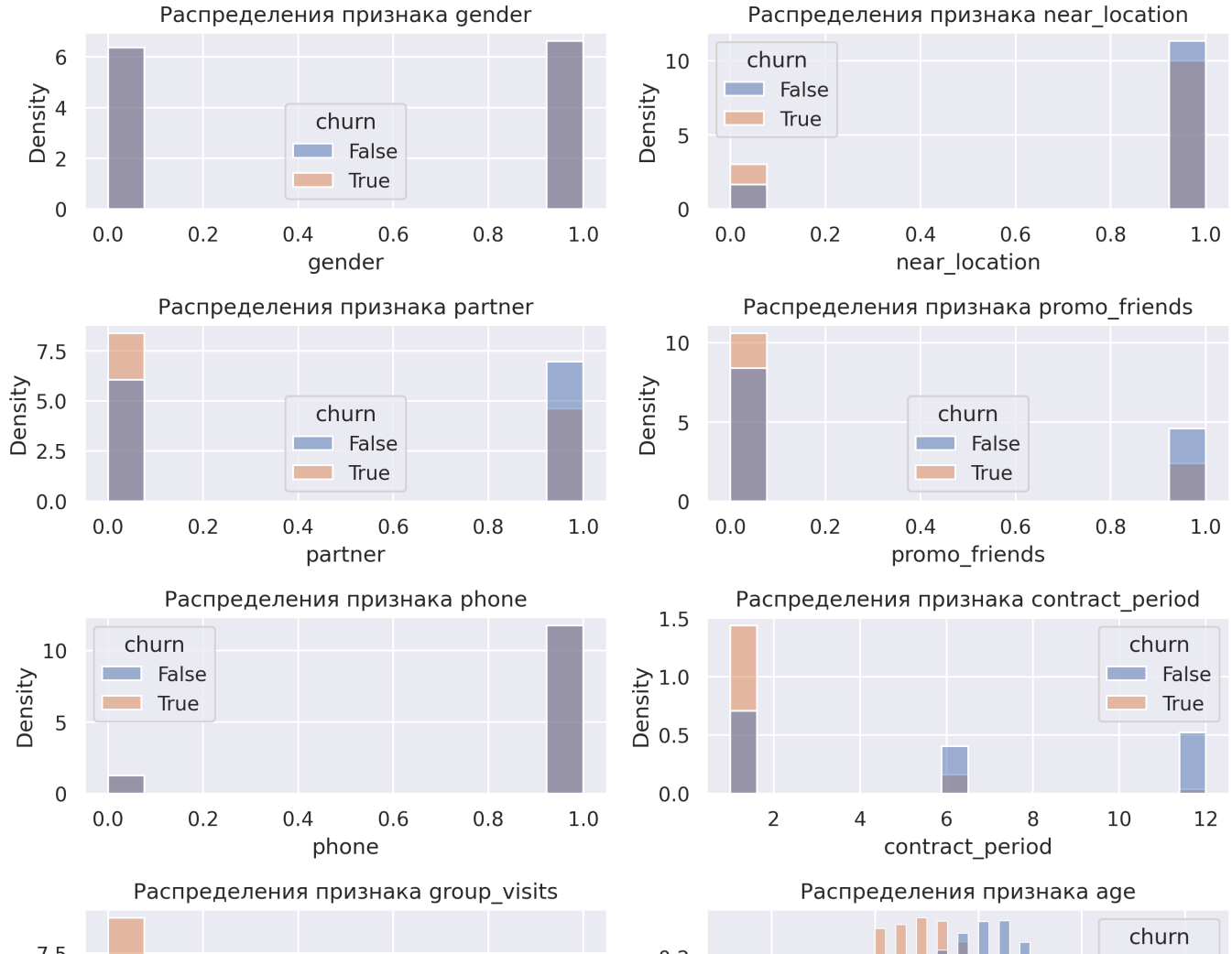

lifetime
остались
ушли
Количество клиентов
Значение признака

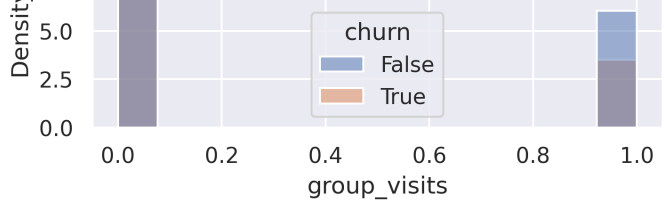

avg_class_frequency_total
остались
ушли
Количество клиентов
Значение признака

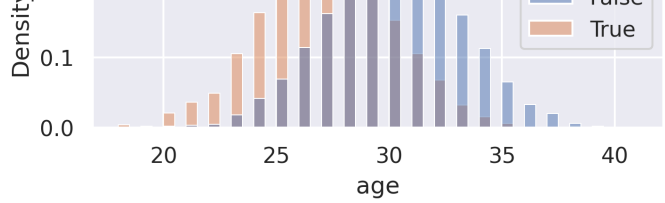


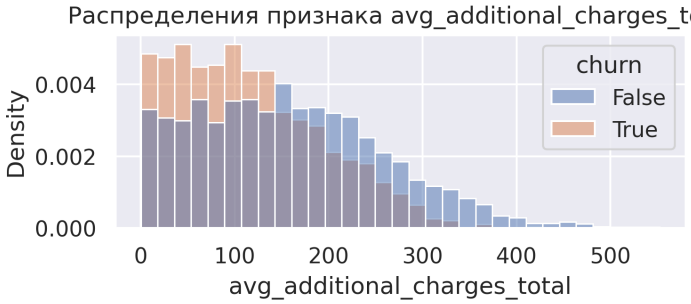
In [12]:

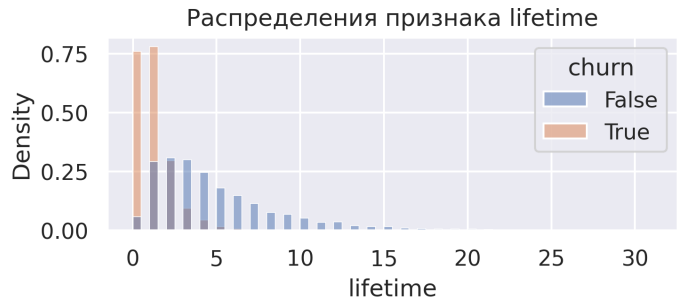
```
fig, axs = plt.subplots(len(df.columns) // 2, 2)
fig.set_size_inches(10, 16)
fig.set_dpi(300)

new_axs = [item for sublist in axs for item in sublist]

for i, column in enumerate(df.columns):
    sns.histplot(data=df, x=column, hue='churn',  stat='density', common_norm=False,  ax=n
    new_axs[i].set_title('Распределения признака {}'.format(column))

plt.tight_layout()
plt.show()
```

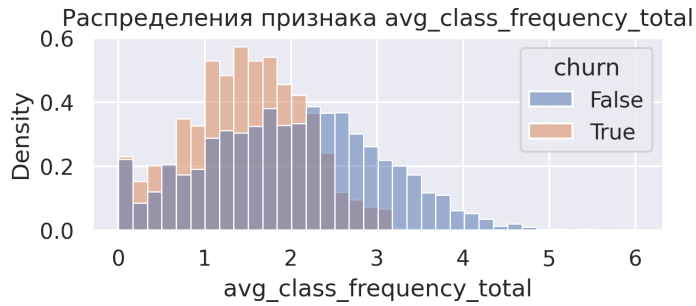


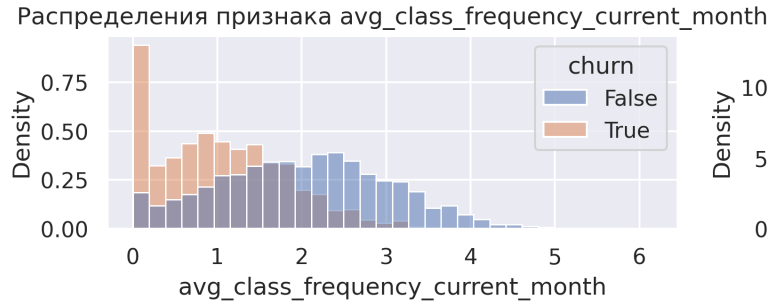


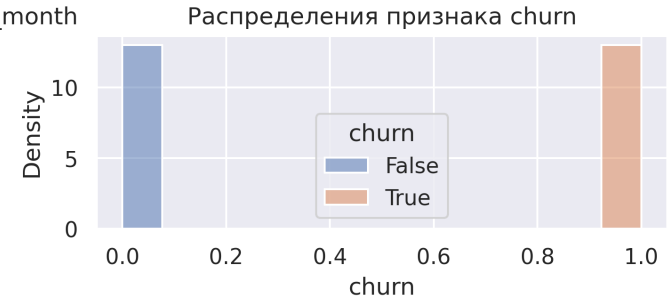












Выделим основные различия, которые видим на основании графиков меньший отток демонстрируют:

- сотрудники компаний партнеров;
- клиенты, проживающие в районе фитнес центра;
- клиенты с длительным сроком действия абонемента;
- клиенты посещающие групповые занятия, при этом для не посещающих разницы в оттоке не наблюдается;
- клиенты в возрасте от 28 лет

Не проявляют лояльность и чаще попадают в отток

- клиенты по акции "приведи друга"
- клиенты со сроком действия абонемента менее 6 месяцев, в особенности краткосрочных 1 месяц;
- клиенты в возрасте до 28 лет;
- те, кто не тратит либо тратит мало на доп. услуги;
- клиенты, у которых остается малый срок до конца срока действия абонемента (или опять же в принципе коротки абонемент);
- клиенты посещающие занятия реже двух раз в неделю

Не наблюдается разницы и отток сопоставим

- между мужчинами и женщинами;

- у не посещающих групповые занятия

В большинстве случаев распределения можно считать нормальными. За исключением, пожалуй, распределений по продаже доп. услуг, времени с момента первого обращения и среднего числа посещений за последний месяц.

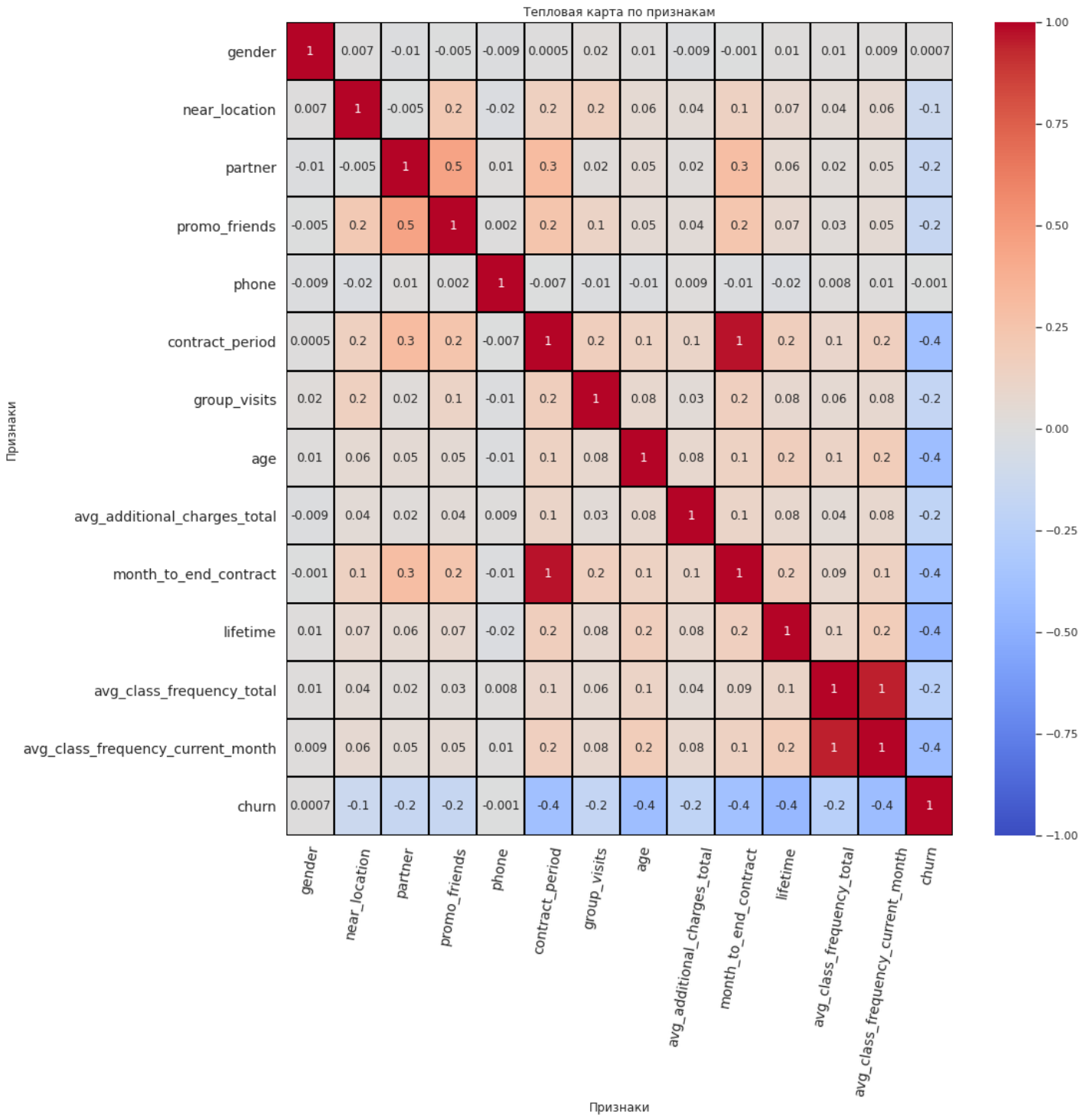
Далее посмотрим данные матрицы корреляций

In [13]:
```
df.corr()
```

Out[13]:

	gender	near_location	partner	promo_friends	phone	contract_period	gr
gender	1.000000	0.006699	-0.010463	-0.005033	-0.008542	0.000502	
near_location	0.006699	1.000000	-0.005119	0.210964	-0.015763	0.150233	
partner	-0.010463	-0.005119	1.000000	0.451960	0.009970	0.306166	
promo_friends	-0.005033	0.210964	0.451960	1.000000	0.001982	0.244552	
phone	-0.008542	-0.015763	0.009970	0.001982	1.000000	-0.006893	
contract_period	0.000502	0.150233	0.306166	0.244552	-0.006893	1.000000	
group_visits	0.017879	0.154728	0.022710	0.120170	-0.010099	0.169991	
age	0.013807	0.058358	0.047480	0.050113	-0.011403	0.138249	
avg_additional_charges_total	-0.009334	0.040761	0.022941	0.036898	0.009279	0.111445	
month_to_end_contract	-0.001281	0.143961	0.294632	0.239553	-0.011196	0.973064	
lifetime	0.013579	0.070921	0.061229	0.072721	-0.018801	0.170725	
avg_class_frequency_total	0.014620	0.043127	0.024938	0.028063	0.008340	0.096211	
avg_class_frequency_current_month	0.009156	0.062664	0.045561	0.053768	0.013375	0.159407	
churn	0.000708	-0.128098	-0.157986	-0.162233	-0.001177	-0.389984	

In [14]:
```
plt.figure(figsize=(15, 15))
sns.heatmap(df.corr(), annot = True, fmt='.1g', vmin=-1, vmax=1, center= 0, cmap= 'coolwa
plt.title('Тепловая карта по признакам')
plt.xticks(fontsize=14, rotation=80)
plt.yticks(fontsize=14)
plt.ylabel('Признаки')
plt.xlabel('Признаки');
```



- явной зависимости факта оттока от признаков не обнаружено;
- присутствует линейная зависимость между признаками - `contract_period` и `month_to_end_contract`, `avg_class_frequency_current_month` и `avg_class_frequency_total`, что само по себе логично и очевидно;
- большинство признаков не связаны друг с другом;

Поскольку для построения модели прогнозирования не желательно наличие сильно скоррелированных показателей, для построения модели прогнозирования оттока решаю убрать из выборки такие признаки.

In [15]:
```
df_for_model = df.drop('avg_class_frequency_total', axis=1)
df_for_model = df.drop('month_to_end_contract', axis=1)
#df = df.drop('contract_period', axis=1)
#df = df.drop('avg_class_frequency_current_month', axis=1)
```

Вывод:

На данном шаге мы провели предобработку данных и исследовательский анализ EDA. Выявили для каких клиентов в большей степени характерен отток.

- пропусков не выявлено,
- категориальные переменные уже имеют числовой вид
- нет признаков, коррелирующих с целевой переменной;
- обнаружены признаки сильно коррелирующие друг с другом, для построения модели прогнозирования их использовать не будем.

Шаг. Построение модели прогнозирования оттока пользователей

Выделение обучающей и валидационной выборок

In [16]:
```
# разделим наши данные на признаки (матрица X) и целевую переменную (y)
X = df_for_model.drop('churn', axis=1)
y = df_for_model['churn']

# разделяем модель на обучающую и валидационную выборку
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

Стандартизация данных

In [17]:
```
# создадим объект класса StandardScaler и применим его к обучающей выборке
scaler = StandardScaler()
X_train_st = scaler.fit_transform(X_train) #обучаем scaler и одновременно трансформируем м
#применяем стандартизацию к матрице признаков для тестовой выборки
X_test_st = scaler.transform(X_test)
```

Обучение модели на train-выборке двумя способами: логистической регрессией, случайным лесом.

In [18]:
```
# задаем алгоритм для нашей модели, сначала Логистическая регрессия
model_LR = LogisticRegression(solver='liblinear', random_state=0)
# обучение модели
model_LR.fit(X_train_st, y_train)
# воспользуйтесь уже обученной моделью, чтобы сделать прогнозы
predictions_LR =  model_LR.predict(X_test_st)
probabilities_LR = model_LR.predict_proba(X_test_st)[:,1]

model_RF = RandomForestClassifier(random_state=0)
model_RF.fit(X_train_st, y_train)
predictions_RF =  model_RF.predict(X_test_st)
probabilities_RF = model_RF.predict_proba(X_test_st)[:,1]
```

Оценка метрики accuracy, precision и recall для обеих моделей на валидационной выборке. Сравнение по ним модели.

In [19]:
```
print ('Значениея метрик для модели логистической регресси')
print ('accuracy = ', round(accuracy_score(y_test, predictions_LR), 2))
print ('precision = ', round(precision_score(y_test, predictions_LR), 2))
print ('recall = ', round(recall_score(y_test, predictions_LR), 2))
```

```
Значениея метрик для модели логистической регресси
accuracy =  0.92
precision =  0.85
recall =  0.83
```

`accuracy/точность` : Доля правильных ответов на уровне 0.92, отличный показатель.

Метрики без привязки к балансу "классов" также показывают неплохие результаты

`precision/точность` (доля правильных ответов только среди целевого класса) 0,85

`recall/полнота` (сколько реальных объектов класса мы смогли обнаружить с помощью модели) 0,83

In [20]:
```
print ('Значениея метрик для модели случайный лес')
print ('accuracy = ', round(accuracy_score(y_test, predictions_RF), 2))
print ('precision = ', round(precision_score(y_test, predictions_RF), 2))
print ('recall = ', round(recall_score(y_test, predictions_RF), 2))
```

```
Значениея метрик для модели случайный лес
accuracy =  0.91
precision =  0.83
recall =  0.81
```

В данной модели значение аналогичных метрик сопоставимы и уступают логистической регрессии на 1-2 сотых.

Вывод:

На данном было построено две модели прогнозирования оттока клиентов.

- логистическая регрессия;
- случайный лес

и оценены их качество для наших данных с помощью метрик accuracy (доля правильных ответов), precision (точность), recall (полнота). По всем метрикам получаем не плохие значения у обоих моделей, но чуть лучше демонстрирует себя *Логистическая регрессия* .

На основании полученных значений метрик, можно сказать, что обе модели показывают себя неплохо и способны делать приемлемые прогнозы.

Шаг. Кластеризация клиентов.

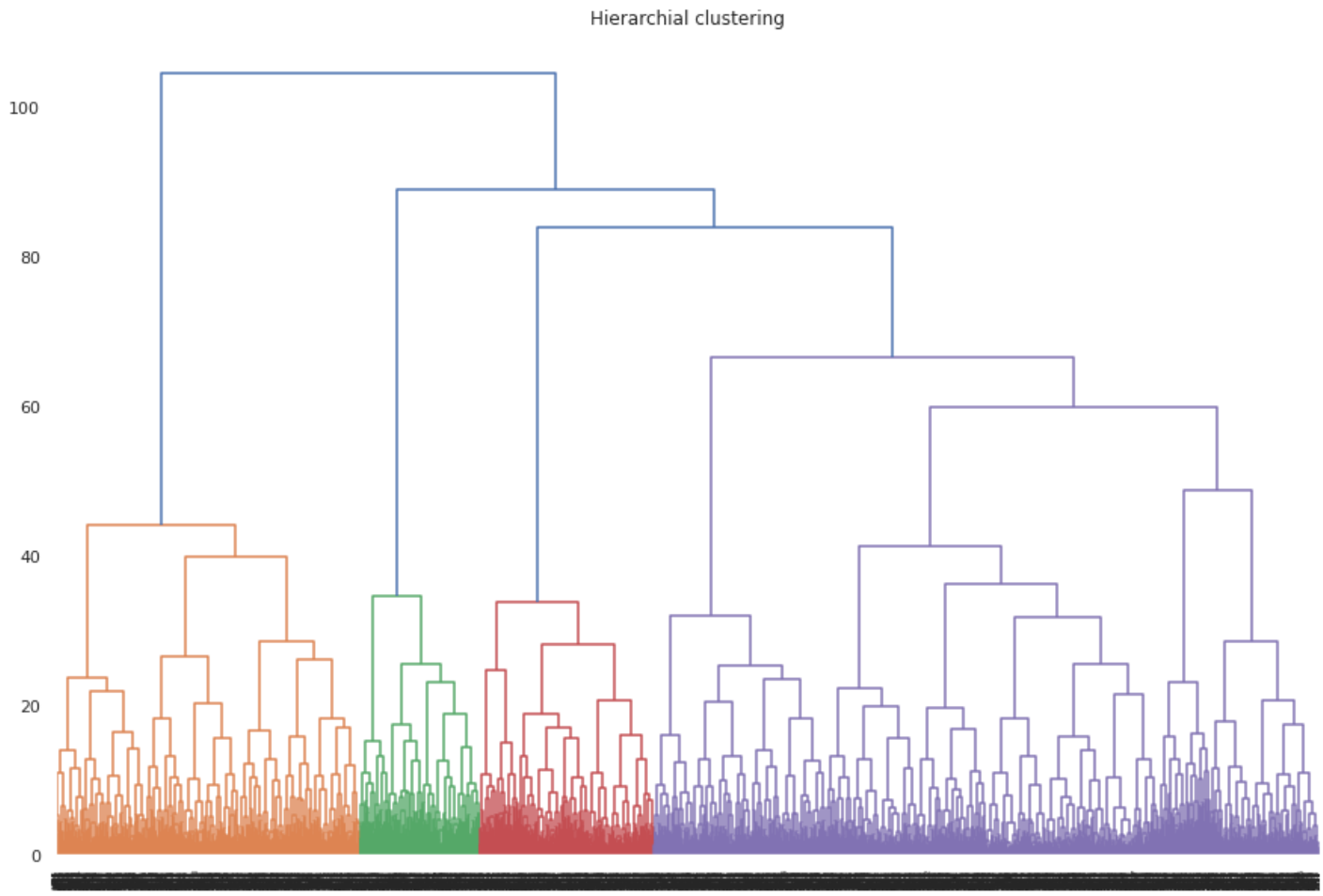
Для начала отбросим столбец с оттоком и стандартизируем данные.

In [21]:
```
X = df.drop('churn', axis=1)
```

In [22]:
```
# стандартизируем данные
sc = StandardScaler()
X_sc = sc.fit_transform(X)

# в переменной linked сохранена таблица «связок» между объектами, её можно визуализировать
linked = linkage(X_sc, method = 'ward')
```

In [23]:
```
# строим дендрограмму
plt.figure(figsize=(15, 10))
dendrogram(linked, orientation='top')
```

```
plt.title('Hierarchial clustering')
plt.show()
```



Можем предположить, что данные целесообразно разделить на 4 кластера. Согласно условию задачи, установим число кластеров равным 5 для дальнейшего исследования.

In [24]:
```
km = KMeans(n_clusters=5, random_state=0) # задаём число кластеров, равное 5, и фиксируем
labels = km.fit_predict(X_sc) # применяем алгоритм к данным и формируем вектор кластеров
```

In [25]:
```
# добавляем в исходные данные метки с номерами кластеров
df['cluster_km'] = labels
df.head()
```

Out[25]:

	gender	near_location	partner	promo_friends	phone	contract_period	group_visits	age	avg_additional_charges_
0	True	True	True	True	False	6	True	29	14.22
1	False	True	False	False	True	12	True	31	113.20
2	False	True	True	False	True	1	False	28	129.44
3	False	True	True	True	True	12	True	33	62.66
4	True	True	True	True	True	1	False	26	198.36

Посмотрим средние значения метрик для каждого кластера.

In [26]:
```
round (df.groupby('cluster_km').mean().T, 2)
```

Out[26]:

cluster_km	0	1	2	3	4

cluster_km	0	1	2	3	4
gender	0.50	0.52	0.50	0.49	0.56
near_location	0.96	0.86	0.00	1.00	0.98
partner	0.78	0.47	0.46	0.35	0.36
promo_friends	0.57	0.31	0.08	0.24	0.23
phone	1.00	0.00	1.00	1.00	1.00
contract_period	10.89	4.79	2.35	1.95	2.67
group_visits	0.54	0.43	0.22	0.34	0.47
age	29.98	29.30	28.48	28.17	30.13
avg_additional_charges_total	160.76	143.96	135.46	131.62	161.66
month_to_end_contract	9.95	4.48	2.20	1.86	2.46
lifetime	4.74	3.92	2.81	2.44	4.90
avg_class_frequency_total	1.98	1.85	1.66	1.25	2.85
avg_class_frequency_current_month	1.97	1.72	1.48	1.01	2.85
churn	0.03	0.27	0.44	0.51	0.07

Посмотрим сколько клиентов попало в каждый кластер

In [27]:
```
df.groupby('cluster_km')['gender'].count()
```

Out[27]:
```
cluster_km
0    1010
1     385
2     505
3    1262
4     838
Name: gender, dtype: int64
```

Получили кластеры размерностью от 385 до 1262. Оценим качество проведенной кластеризациии с помощью метрики `Silhouette_score`

In [28]:
```
print('Silhouette_score: = ', silhouette_score(X_sc, labels))
```

```
Silhouette_score: =  0.14140953623023353
```

Получаем значение метрики на уровне 0,14, это означает, что возможна лучшая кластеризация чем у нас получилась.

Самые лояльные клиенты попали в кластер 0 и кластер 4 (отток на уровне от 3 до 7 %), худшие показатели оттока в кластерах 2 и 3 (44% и 51%) соответственно.

Можно отметить, что в группу с минимальным оттоком в основном попали клиенты со средним возрастом 30 лет, проживающие рядом с фитнес центром, тратящие в среднем больше остальных на доп. услуги. Примерно половина из них посещает групповые занятия, многие (около 78% являются сотрудниками компаний - партнеров. Посещают зал эти клиенты 2 раза в неделю. Являются владельцами годовых абонементов и прозанимались 2 месяца из них.

В кластере где собралось много отточных клиентов относительно мало участников партнерских и "дружеских" программ, более молодые клиенты (около 28), клиенты с краткосрочным абонементом реже

остальных посещающие зал.

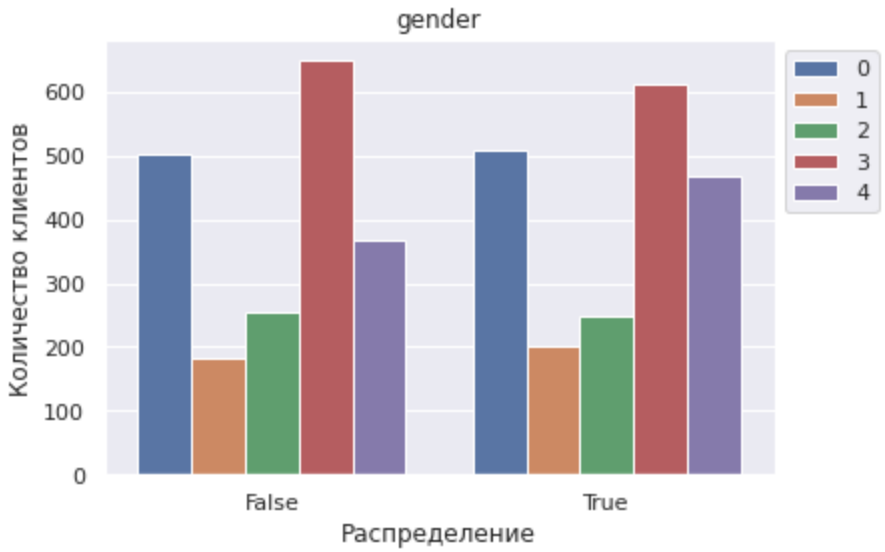
Полученные средние в разбивке по кластерам значения не противоречат сделанным ранее наблюдениям.

На данном шаге можно предположить, что и по итогам исследования кластер 0 покажет лучшие результаты относительно общего оттока в выборке и его метрики будут основными для построения портрета нашего клиента.

Распределение признаков по кластерам

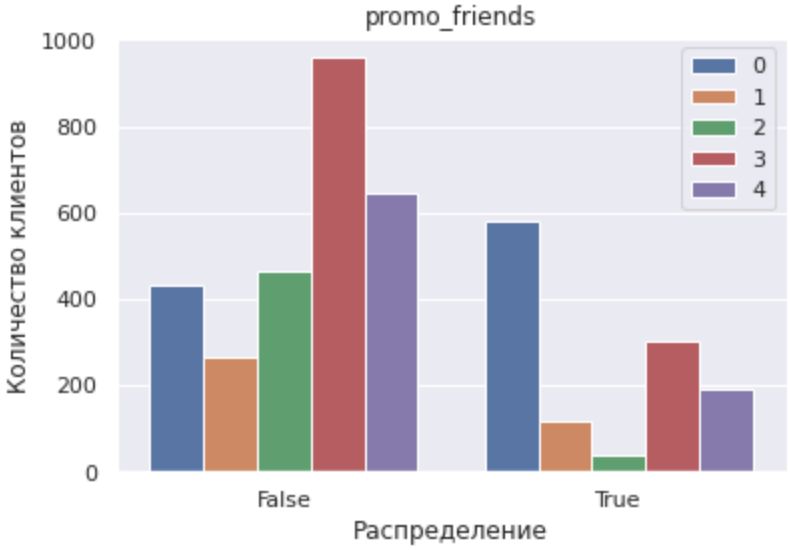
Далее построим распределения признаков по кластерам. Достаточно наглядно на мой взгляд для булевых значений и срока абонемента распределения иллюстрируют столбчатые графики, для остальных величин boxplot.

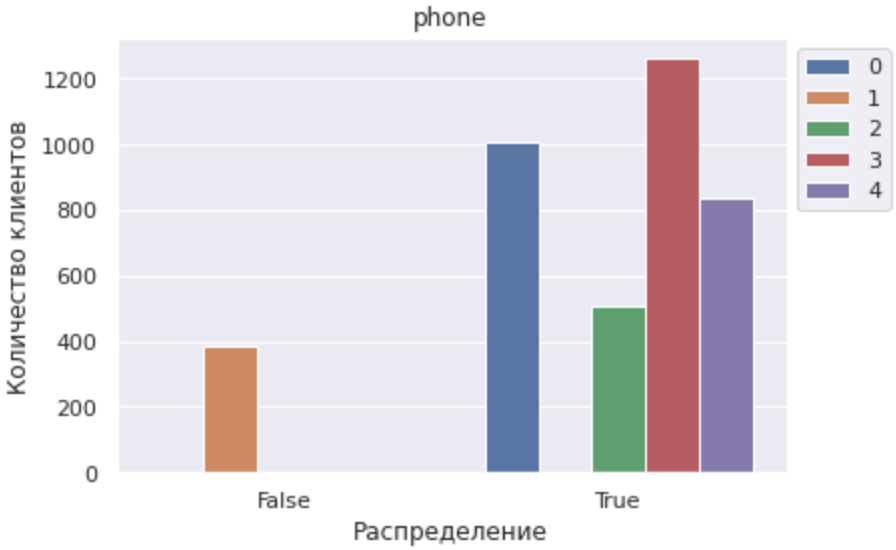
In [29]:
```
# Построим график для логических и дискретных величин
values = ['gender', 'near_location', 'partner', 'promo_friends', 'phone', 'contract_perioc
'group_visits', 'churn']
fig, ax = plt.subplots()
for column in values:
    plt.title(column)
    sns.countplot(data=df, x=column, hue='cluster_km')
    plt.xlabel('Распределение')
    plt.ylabel('Количество клиентов')
    plt.legend(bbox_to_anchor=(1, 1))
    plt.show()
```

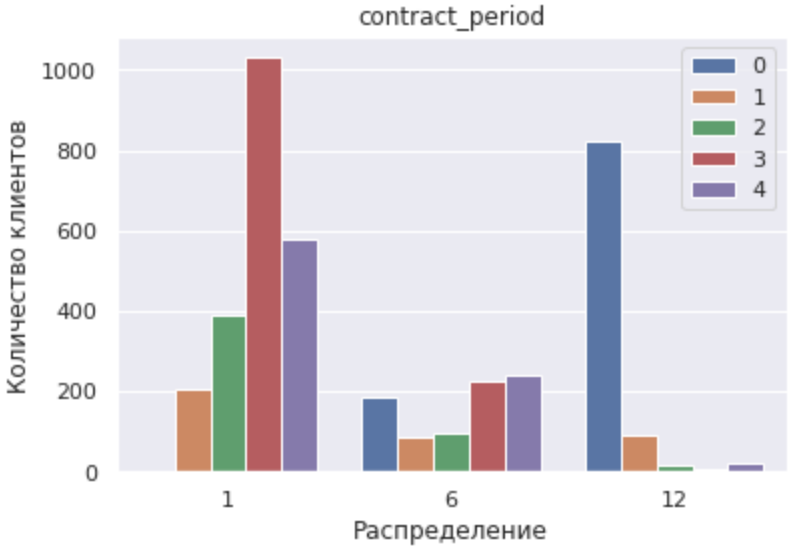


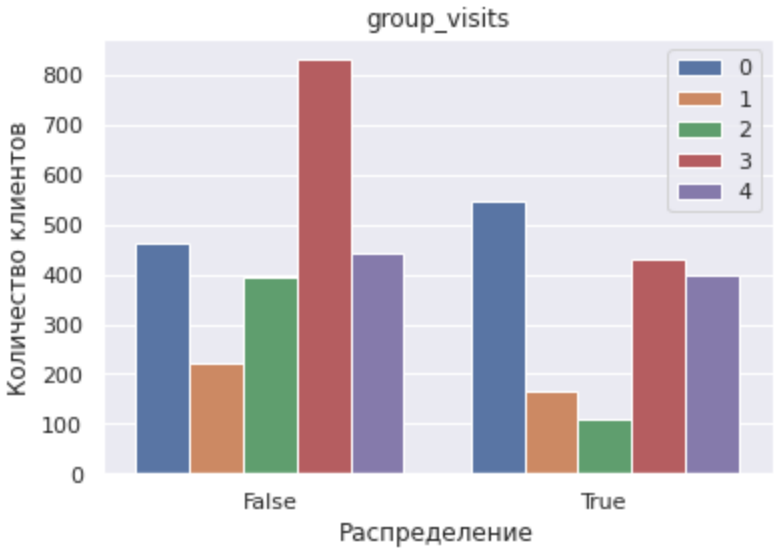

near_location
Количество клиентов
0
200
400
600
800
1000
1200
0
1
2
3
4
False
True
Распределение

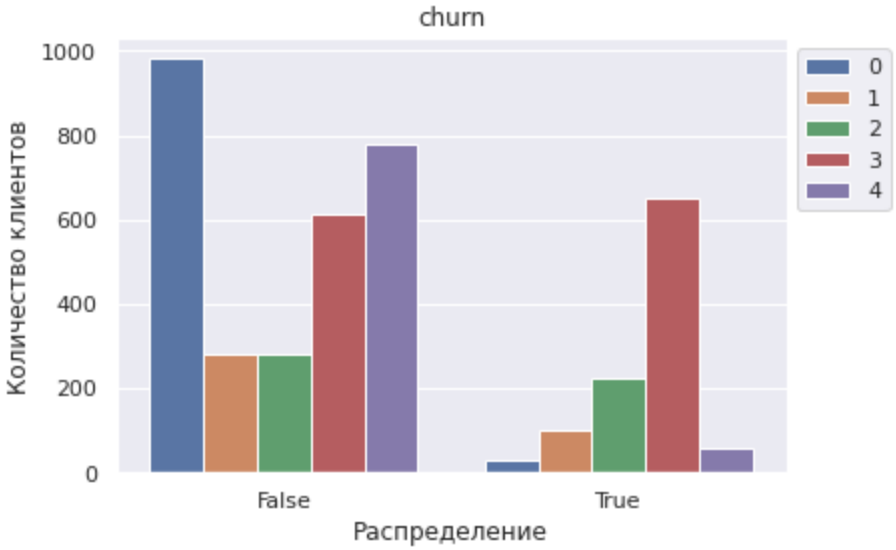

partner
Количество клиентов
0
100
200
300
400
500
600
700
800
0
1
2
3
4
False
True
Распределение


promo_friends
Количество клиентов
0
200
400
600
800
1000
0
1
2
3
4
False
True
Распределение


phone
Количество клиентов
0
200
400
600
800
1000
1200
False
True
Распределение
0
1
2
3
4

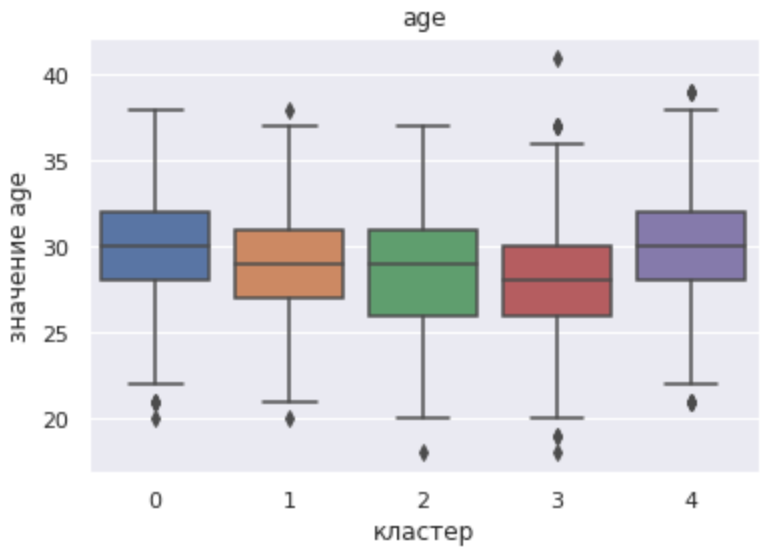

contract_period
Количество клиентов
0
200
400
600
800
1000
1
6
12
Распределение
0
1
2
3
4

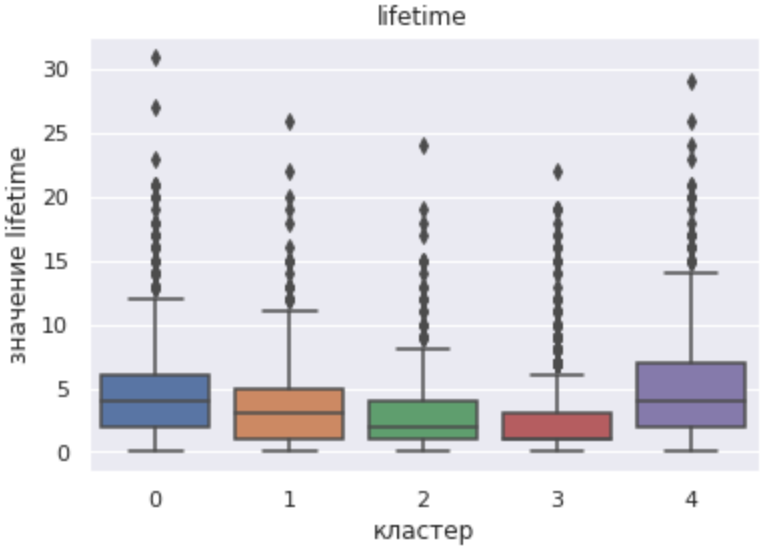

group_visits
Количество клиентов
0
100
200
300
400
500
600
700
800
False
True
Распределение
0
1
2
3
4

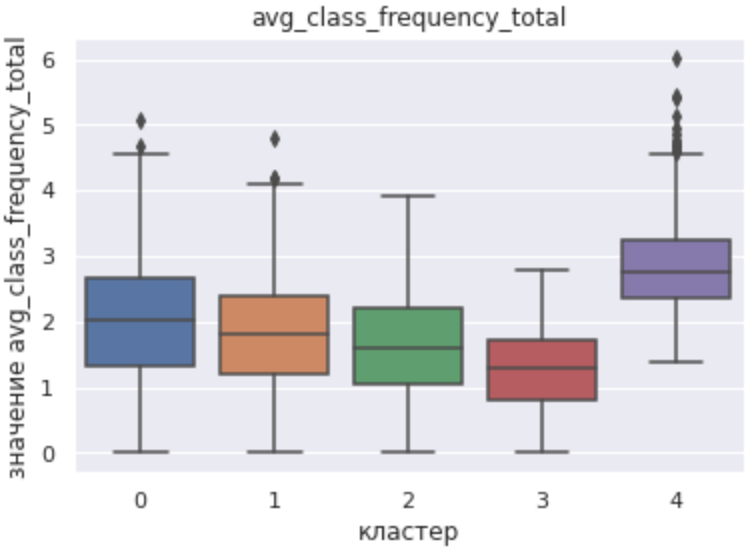


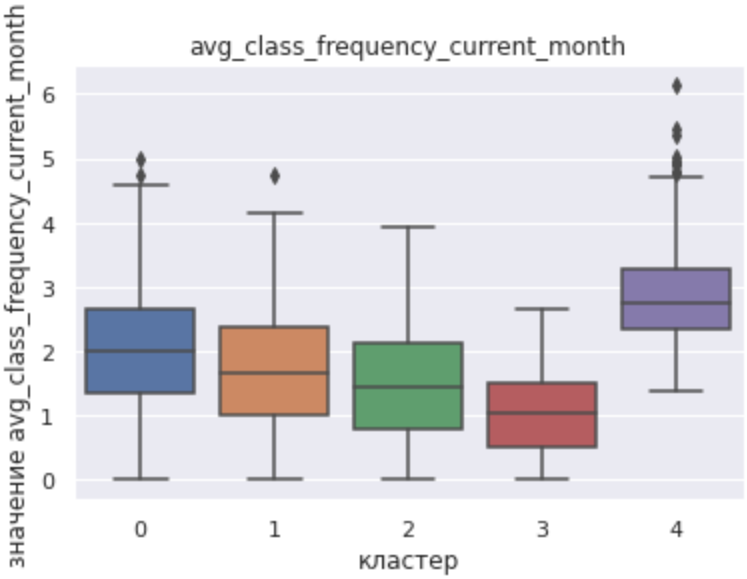
In [30]:

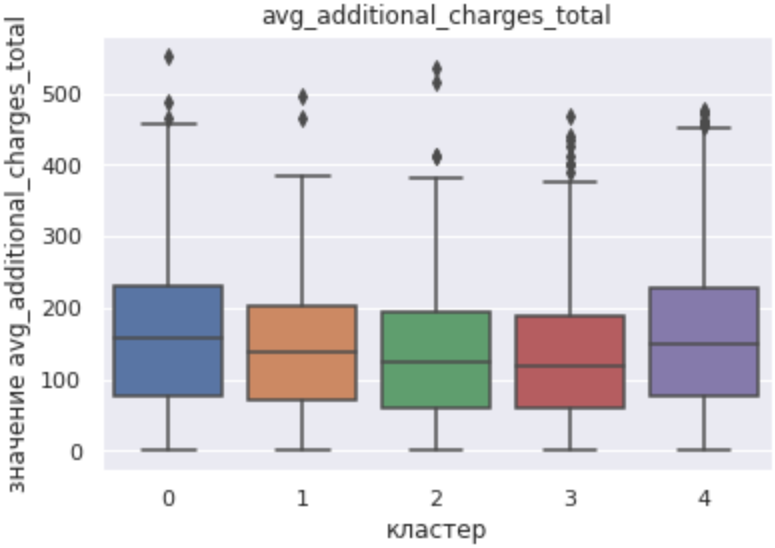
```
values = ['age', 'lifetime', 'avg_class_frequency_total', 'avg_class_frequency_current_mon
          'month_to_end_contract']
fig, ax = plt.subplots()
for column in values:
    plt.title(column)
    sns.boxplot(data=df, x='cluster_km', y=column)
    plt.xlabel('кластер')
    plt.ylabel('значение '+column)
    plt.show()
```

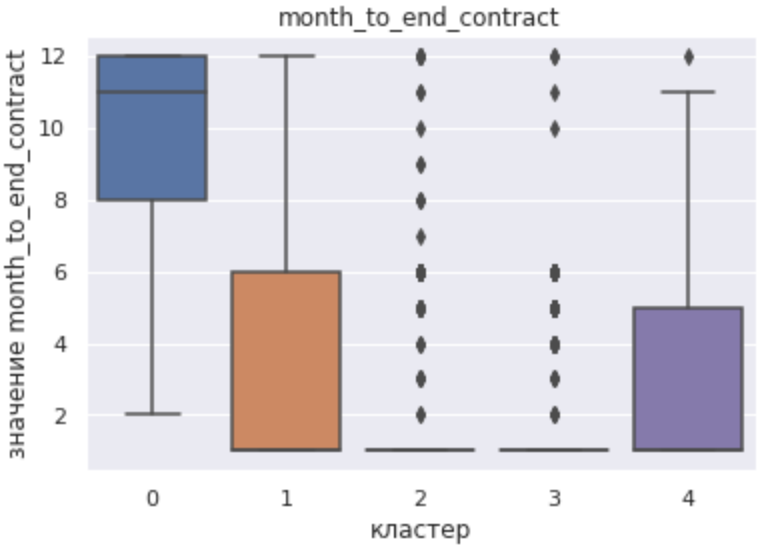





avg_class_frequency_total
значение avg_class_frequency_total
0
1
2
3
4
5
6
0
1
2
3
4
кластер


avg_class_frequency_current_month
значение avg_class_frequency_current_month
0
1
2
3
4
5
6
0
1
2
3
4
кластер


avg_additional_charges_total
значение avg_additional_charges_total
0
100
200
300
400
500
0
1
2
3
4
кластер



Для всех кластеров количество мужчин и женщин разделилось почти поровну. Акционные клиенты (партнеры, "приведи друга") менее отточны за исключением кластера 0.

Выделим основные маркеры характерные для наших кластеров по итогам построения графиков распределения:

Кластер 0:

В кластер 0 попали преимущественно следующие клиенты

- проживающие рядом с фитнес - центром;
- сотрудники компаний-партнеров;
- имеющие длительные годовые абонементы;
- возраст около 30;
- относительно много тратящие на доп. услуги;
- посещающие занятия в среднем дважды в неделю;
- чуть чаще посещающие групповые занятия;
- **с самой низкой долей оттока.**

Кластер 1:

В кластер 1 попали преимущественно следующие клиенты

- проживающие не далеко от фитнес - центра
- доля сотрудников партнеров и остальных почти поровну и невысокая доля пришедших по рекомендации;
- не предоставившие номер телефона;
- срок абонемента чаще всего 1 месяц;
- реже посещающие групповые занятия;
- средний возраст ниже 30;
- в среднем реже 2 раз в неделю посещающие занятия;
- **достаточно высокая доля оттока.**

Кластер 2:

В кластер 2 попали преимущественно следующие клиенты

- проживающие далеко от фитнес - центра;

- не по акции "приведи друга";
- короткие абонементы чаще 1 месяц;
- реже посещающие групповые занятия;
- возраст ниже 30;
- в среднем реже 2 раз в неделю посещающие занятия;
- **достаточно высокая доля оттока.**

Кластер 3:

В кластер 3 попали преимущественно следующие клиенты

- проживающие рядом;
- краткосрочные абонементы;
- реже посещающие групповые занятия;
- самые молодые (около 27-28);
- редко бывающие в зале чаще 1 раза в неделю;
- меньшая часть в этом кластере участники партнерской либо "дружеской" акций;
- **самая высокая доля оттока.**

Кластер 4:

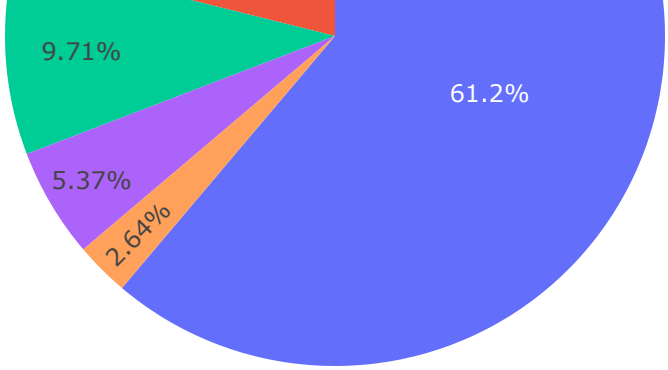
В кластер 4 попали преимущественно следующие клиенты

- проживающие рядом;
- возраст около 30;
- в среднем чаще 2 раз в неделю посещающие клуб;
- активно покупающие доп. услуги;
- со средним либо краткосрочным абонементом;
- меньшая часть в этом кластере участники партнерской либо "дружеской" акций;
- **не высокая доля оттока**

Доли оттока по кластерам

Теперь посчитаем долю оттока в каждом кластере относительно общего числа оттоков в выборке.

In [31]:
```
part_churn = df.query('churn==1').groupby('cluster_km')['churn'].count() / df.query('churr
fig = go.Figure(data=[go.Pie(labels=part_churn.index, values=part_churn.values, title='Дол
fig.show() ;
```



In [32]:
```
part_churn
```

Out[32]:
```
cluster_km
0    0.026390
1    0.097078
2    0.211122
3    0.611687
4    0.053723
Name: churn, dtype: float64
```

Вывод

Как и предполагалось `Кластер 0` показывает лучшие результаты по оттоку клиентов 2,64%, не плохие показатели у `кластера 4` на уровне 5,37%. Ключевые черты характерные для этих кластеров можно использовать для построение портрета нашего "идеального" лояльного клиента.

Это человек в возрасте около 30 лет, проживающий рядом, посещающий зал 2 иногда 3 раза в неделю, любитель групповых занятий и пользующийся доп услугами, владелец годового абонемента, чаще всего сотрудник компании-партнера.

Худшие показатели у кластеров 3 и 2 (61,2% и 21,1% соответственно), аналогично ключевые черты характерные для этих кластеров можно использовать для построение портрета нашего потенциально отточного клиента.

Это человек моложе 30 лет, проживающий далеко от клуба, посещающий клуб не чаще раза в неделю, владелец краткосрочного абонемента, не склонный к групповым занятиям.

Общий вывод

В данной работе мы проанализировали данные клиентов фитнес центра. На основании имеющихся данных было построено две модели прогнозирования оттока клиентов.

- логистическая регрессия;
- случайный лес

Основываясь на показаниях ключевых метрик оценки моделей accurancy, precision и recall можно сказать, что обе модели хорошо показывает себя в прогнозировании оттока. По совокупности чуть лучше показывает себя модель *Логистическая регрессия* .

Далее была проведена кластеризация клиентов. Построенная дендрограмма показала 4 явных кластера,

согласно заданию, выборку разделили на 5 кластеров. Построили распределение признаков по кластерам, на основании которых построили портрет лояльного и ненадежного клиента.

Основные числовые характеристики и портреты даны по ходу исследования. Здесь сфокусируемся на рекомендациях отделу маркетинга.

Рекомендации отделу маркетинга

Проводить маркетинговые активности, направленные на целевую аудиторию:

- акцент на рекламу в близлежащих районах, клиенты из далека не станут постоянными;
- реклама должна нести в себе ценности близкие и понятные человеку в возрасте 30+ (например, стабильность, забота о здоровье, восстановление/поддержание хорошей фигуры/формы, высокое качество тренажёров/услуг/программ и т.п.);

Работа с действующими клиентами:

- мотивировать действующих клиентов на регулярные (2-3 раза в неделю) посещения (акцент в рекламе, что достижение целей о здоровье/фигуре и т.п. возможно только при такой интенсивности посещений), возможно добавить бесплатную для клиента услугу напоминания о тренировках по CMC/WA/Telegram и т.п.
- активнее предлагать и по возможности разнообразить групповые тренировки, клиент, регулярно занимающийся в группе и приходить будет чаще и менее склонен к оттоку, вероятно за счет так называемого "стадного" инстинкта либо за счет поддерживаемого разнообразием интереса
- разнообразить и увеличить число предложений дополнительных платных услуг.

Тарифная политика:

- мотивировать отдел продаж предлагать и продавать клиентам более долгосрочные абонементы (более высокие бонусы от продаж таких абонементов и/или скидка клиенту на долгосрочный абонемент растущая со сроком);

Работа со склонными к оттоку клиентами:

- поскольку как правило краткосрочный абонемент дороже, в его стоимость можно включить ряд персональных тренировок, предусматривающих составление программы занятий;

Акции:

- не прекращать и наращивать партнерскую программу в организациях, т.к. клиенты из компаний партнеров менее склонны к оттоку;
- акция "приведи друга" также показывает себя хорошо, рекомендация продолжать.