

GeekBrains Базы данных для аналитиков Урок 4. Типовая аналитика маркетинговой активности

1. Определяем критерии для каждой буквы R, F, M (т.е. к примеру, R – 3 для клиентов, которые покупали ≤ 30 дней от последней даты в базе, R – 2 для клиентов, которые покупали > 30 и менее 60 дней от последней даты в базе и т.д.)
2. Для каждого пользователя получаем набор из 3 цифр (от 111 до 333, где 333 – самые классные пользователи)
3. Вводим группировку, к примеру, 333 и 233 – это Vip, 1XX – это Lost, остальные Regular (можете ввести более глубокую сегментацию)
4. Для каждой группы из п. 3 находим кол-во пользователей, кот. попали в них и % товарооборота, которое они сделали на эти 2 года.
5. Проверяем, что общее кол-во пользователей бьется с суммой кол-во пользователей по группам из п. 3 (если у вас есть логические ошибки в создании групп, у вас не сойдутся цифры). То же самое делаем и по деньгам.

Recency (давность) — давность сделки, чем меньше времени прошло с момента последней активности клиента, тем больше вероятность, что он повторит действие Frequency (частота) — количество сделок, чем больше каких-либо действий совершил клиент, тем больше вероятность того, что он его повторит в будущем Monetary (деньги) — сумма сделок, чем больше денег было потрачено, тем больше вероятность того, что он сделает заказ

In [1]:

```
import pandas as pd
from datetime import datetime

from datetime import date
```

Загружаем данные из файла и преобразуем стоимость заказа и дату заказа

In [2]:

```
df = pd.read_csv('orders_20190822.csv', sep = ';')
df['price'] = pd.to_numeric(df['price'].str.replace(',', '.'),
downcast="integer")
df['o_date'] = pd.to_datetime(df['o_date'])
```

Группируем таблицу заказов по покупателям + для каждой группы считаем количество покупок и сумму покупок считаем количество дней, прошедших от текущей даты до последней покупки

In [3]:

```
s1 = df.groupby('user_id')['price'].sum()
s2 = df.groupby('user_id')['price'].count()
s3 = df.groupby('user_id')['o_date'].max()
df1 = pd.merge(s1, s2, how='inner', on='user_id')
df2 = pd.merge(df1, s3, how='inner', on='user_id')
df2['user_id'] = df2.index
df2.rename(columns = {'price_x':'sum_price', 'price_y':'purchases',
'o_date':'last_o_date'}, inplace = True)
df = df2[['user_id', 'sum_price', 'purchases', 'last_o_date']]
df = df.copy()
df['curr_days']=df['last_o_date'].apply(lambda x: (datetime.now() - x).days)
```

анализируем значения выборки для выборов критериев группировки

In [4]:
df.describe()
Out[4]:

	user_id	sum_price	purchases	curr_days
count	1.015119e+06	1.015119e+06	1.015119e+06	1.015119e+06
mean	3.060034e+06	4.475029e+03	1.972975e+00	1.631339e+03
std	1.601093e+06	2.827953e+04	9.461065e+00	2.067240e+02
min	0.000000e+00	-1.848000e+02	1.000000e+00	1.357000e+03
25%	1.707710e+06	9.303000e+02	1.000000e+00	1.438000e+03
50%	2.993663e+06	1.840300e+03	1.000000e+00	1.602000e+03
75%	4.425940e+06	3.963050e+03	1.000000e+00	1.779000e+03
max	5.919156e+06	1.173418e+07	3.183000e+03	2.087000e+03

Задаем критерии для группировки

In [5]:
r1 = df.curr_days.quantile(0.2)
r2 = df.curr_days.quantile(0.75)
f1 = df.purchases.quantile(0.8)
f2 = df.purchases.quantile(0.9)
m1 = df.sum_price.quantile(0.35)
m2 = df.sum_price.quantile(0.85)
In [6]:
df['Recency']=df['curr_days'].apply(lambda x: 3 if x < r1 else 2 if x < r2
else 1)
In [7]:
df['Frequency']=df['purchases'].apply(lambda x: 1 if x < f1 else 2 if x < f1
else 3)
In [8]:
df['Monetary']=df['sum_price'].apply(lambda x: 1 if x < m1 else 2 if x < m2
else 3)

Группируем пользователей:

In [9]:
df['RFM_sum'] = df.apply(lambda row: row.Recency + row.Frequency +
row.Monetary, axis = 1)
In [10]:
df
Out[10]:

	user_id	sum_price	purchases	last_o_date	curr_days	Recency	Frequency	Monetary	RFM
user_id									
0	0	494.2	1	2016-08-12	1863	1	1	1	3
1	1	13844.6	2	2017-01-08	1714	2	3	3	8
76	76	1863.4	2	2017-09-11	1468	2	3	2	7
90	90	6561.8	4	2017-12-20	1368	3	3	3	9
91	91	5413.1	4	2017-11-10	1408	3	3	2	8
...
5919114	5919114	1272.6	1	2017-12-31	1357	3	1	2	6

	user_id	sum_price	purchases	last_o_date	curr_days	Recency	Frequency	Monetary	RFM
user_id									
5919118	5919118	7242.2	1	2017-12-31	1357	3	1	3	7
5919128	5919128	886.9	1	2017-12-31	1357	3	1	1	5
5919142	5919142	4934.3	1	2017-12-31	1357	3	1	2	6
5919156	5919156	5019.7	1	2017-12-31	1357	3	1	2	6

1015119 rows × 9 columns

In [11]:

```
# Считаем общую сумму товарооборота за 2 года
summ = df['sum_price'].sum().round()
```

In [12]:

```
for x in reversed(range(3,10)):
    s = df.loc[df['RFM_sum'] == x]['sum_price'].sum().round()
    n = df.loc[df['RFM_sum'] == x]['sum_price'].count()
    print ("группа покупателей:", x, "; количество покупателей:", n, "; доля
от товарооборота:", (s/summ).round(3))
группа покупателей: 9 ; количество покупателей: 42241 ; доля от
товарооборота: 0.26
группа покупателей: 8 ; количество покупателей: 85620 ; доля от
товарооборота: 0.244
группа покупателей: 7 ; количество покупателей: 84038 ; доля от
товарооборота: 0.101
группа покупателей: 6 ; количество покупателей: 122414 ; доля от
товарооборота: 0.125
группа покупателей: 5 ; количество покупателей: 290106 ; доля от
товарооборота: 0.167
группа покупателей: 4 ; количество покупателей: 288408 ; доля от
товарооборота: 0.088
группа покупателей: 3 ; количество покупателей: 102292 ; доля от
товарооборота: 0.015
```