

Тестовое задание Бересневой-Батт Валерии на позицию аналитика данных в [REDACTED]

Примерное время выполнения: 8 часов.

1. Теоретическая часть

1.1. Что такое LTV?

LTV (lifetime value), или "ценность клиента" - сумма денег, которую один клиент в среднем приносит со всех своих покупок.

$LTV = V / n$, где

V - общая выручка на текущий день (выручка с накоплением на каждый день лайфтайма),

n - размер когорты (количество пользователей, пришедших в определенный день).

Если у нас есть данные за первый месяц жизни приложения, как бы вы рассчитали LTV?

Есть данные за месяц по новым клиентам. В таком случае лучше разделить клиентов на когорты по дате прихода в приложение и взять для них выручку с накоплением за тот период, в который они должны окупаться, допустим, первую неделю лайфтайма. Так можно будет проанализировать LTV всех новых пользователей, пришедших в первые три недели, за первые 7 дней пользования.

Примерный процесс:

- Для каждого пользователя оставляем действия, совершенные в первые 7 дней лайфтайма, и определяем день лайфтайма, в который совершена покупка (день покупки минус день прихода).
- В сводной таблице определяем индекс как день прихода, что разделит пользователей на когорты, колонки - дни лайфтайма, а значения - выручка, рассчитанная с накоплением для каждой когорты.
- Группируем первоначальные данные по дате прихода, находим количество новых пользователей для каждой даты (когорты) и добавляем к сводной таблице.
- Делим каждую ячейку с выручкой на размер когорты и получаем LTV.

1.2. У вас есть набор данных о времени старта игровых сессий и данные о платежах игроков. В понедельник, прибегают старший геймдизайнер и говорит, что нужно срочно, до завтра, собрать отчёт для совещания с маркетингом, для оценки трафика, закупленного в понедельник, ровно неделю назад. Какие метрики стоит посчитать в первую очередь? Почему?

Если учесть, что нет данных о стоимости рекламных кампаний и о том, откуда пришли пользователи, то будем считать, что все пользователи, которые пришли в прошлый понедельник, пришли из закупленного трафика (если эта информация есть, то, конечно, отберем только нужных пользователей).

Можно посчитать следующие метрики за их первую неделю пользования приложением:

в первую очередь:

- Conversion rate, или конверсия из неплатящих в платящих. Пользователи должны приносить деньги. Если они не переходят в платящих, то реклама привлекает тех, кто не готов платить за пользование приложением.
- Retention rate, или коэффициент удержания. Это покажет, насколько хорошо реклама привлекает пользователей, заинтересованных в приложении.

во вторую очередь:

- Churn rate, или коэффициент отскока. Эта метрика поможет подтвердить то, что покажет удержание. Если отскок с каждым днем будет замедляться, то, вероятнее всего, кампания привлекает заинтересованных пользователей.
- LTV. Посмотрев, приносит ли пользователь деньги, можно посмотреть, сколько он приносит. Если у отдела маркетинга на совещании будет оглашена стоимость закупки трафика, то можно будет определить, окупались ли привлеченные пользователи.

2. Практическая часть

2.1. В игре меняют цену стартер-пака с 1\$ до 2\$. Надо оценить, как изменилась популярность стартер-пака, как первого платежа. Предполагаем, что уменьшится на 10%.

Сколько пользователей, минимум, нужно закупить для A/B теста, если, текущая конверсия в платящие составляет 6%, а стартер-пак, первым платежом, сейчас покупает 80% заплативших?

Гипотеза: "если изменить цену стартер-пака с 1\$ до 2\$, то конверсия в покупку стартер-пака как первую покупку уменьшится на 10%".



Для начала нужно рассчитать конверсию в покупку стартер-пака как первую покупку:

$$6\% \times 80\% = 4.8\%$$

4.8% новых пользователей не просто совершают первую покупку, а покупают стартер-пак.

Теперь воспользуемся [калькулятором Эвана Миллера](#), чтобы рассчитать размер выборок для каждой группы A/B-теста.

Question: How many subjects are needed for an A/B test?

Baseline conversion rate:	<input type="text" value="4.8"/>	%		4.8%	[link]
Minimum Detectable Effect:	<input type="text" value="10"/>	%		4.32% – 5.28%	
<small>The Minimum Detectable Effect is the smallest effect that will be detected (1-β)% of the time.</small>			<small>Conversion rates in the gray area will not be distinguishable from the baseline.</small>		
<input type="radio"/> Absolute <input checked="" type="radio"/> Relative					

Sample size:

31,572

per variation

Statistical power 1-β: 80% Percent of the time the minimum effect size will be detected, assuming it exists

Significance level α: 5% Percent of the time a difference will be detected, assuming one does NOT exist



В итоге, получили, что каждая группа должна включать в себя 31.5 тыс. пользователей, или 63 тыс. пользователей требуется для проведения такого A/B теста.

В то же время, если, например, в день будет проходить 1 тыс. новых пользователей, то тест должен проводиться два месяца, а это довольно долго и очень дорого.

Если взять гипотезу "если изменить цену стартер-пака с 1\$ до 2\$, то среди покупающих пользователей доля тех, кто приобрел стартер-пак как

первую покупку, уменьшится на 10%”, то получается 263 пользователя на группу, а это, наоборот, очень мало.

Question: How many subjects are needed for an A/B test?

Baseline conversion rate:	<input type="text" value="80"/>	%		80%	[link]
Minimum Detectable Effect:	<input type="text" value="10"/>	%		70% – 90%	
<small>The Minimum Detectable Effect is the smallest effect that will be detected (1-β)% of the time.</small>			<input checked="" type="radio"/> Absolute <input type="radio"/> Relative		
<small>Conversion rates in the gray area will not be distinguishable from the baseline.</small>					

Sample size:

263

per variation

Statistical power 1-β: 80% Percent of the time the minimum effect size will be detected, assuming it exists

Significance level α: 5% Percent of the time a difference will be detected, assuming one does NOT exist

В данном случае следует найти баланс между точностью эксперимента и финансовыми возможностями по закупке трафика.

2.2. Необходимо написать SQL-запросы, позволяющие:

Вывести всех игроков, у которых медиа источник 'Facebook'. В случае, если пользователь совершал платежи, также вывести сумму и количество его платежей.

Вывести результат в виде: user_id, сумма всех платежей, число всех платежей.

Примечание: все запросы написаны для DBeaver (SQLite), потому что он установлен у меня на компьютере. Училась я на PostgreSQL.

WITH

/* находим количество покупок для каждого
* пользователя из Facebook */

cnt AS (SELECT user_id,
COUNT(amount) AS cnt_purchases

FROM payment

WHERE user_id in (SELECT user_id
FROM profile
WHERE media_source = 'Facebook')

GROUP BY user_id),

/* находим сумму покупок для каждого пользователя */

total AS (SELECT user_id,
SUM(amount) AS sum_purchases

FROM payment

GROUP BY user_id)

/* объединяем таблицы */

SELECT total.user_id,
total.sum_purchases,
cnt.cnt_purchases

FROM cnt

LEFT JOIN total ON cnt.user_id = total.user_id;

Посчитать накопительный ARPU, по Life Time для игроков, с 1 апреля по 1 мая включительно, для игроков, у которых первая сессия была с 1 по 16 апреля.

Вывести результат в виде: LT (считаем в днях), накопительный ARPU.

WITH

```
/* находим первые сессии пользователей,
* пришедших 01-16.04;
* объединяем с покупками 01.04-01.05 */
```

```
lifetime AS (SELECT p.user_id,
                    p.amount,
                    strftime('%j', p.time) -
                    strftime('%j', f_s.first_dt) AS lt
FROM (SELECT user_id,
             time AS first_dt
FROM session_open
WHERE session_index = 1
AND time BETWEEN '2022-04-01' AND
                '2022-04-16') AS f_s
LEFT JOIN payment AS p ON p.user_id = f_s.user_id
WHERE p.time BETWEEN '2022-04-01' AND '2022-05-01'),
```

```
/* находим количество уникальных активных пользователей
* для каждого дня лайфтайма */
```

```
unique_users AS (SELECT lt,
                        COUNT (DISTINCT user_id) AS cnt_users
FROM (SELECT s.user_id,
             s.time AS first_dt,
             s_o.time AS session_dt,
             strftime('%j', s_o.time) -
             strftime('%j', s.time) AS lt
FROM session_open AS s
LEFT JOIN session_open AS s_o
ON s_o.user_id = s.user_id
WHERE s.session_index = 1
AND s.time BETWEEN '2022-04-01' AND
                  '2022-04-16'
AND s_o.time BETWEEN '2022-04-01' AND
                  '2022-05-01')
GROUP BY lt)
```

```
/* находим накопительный ARPU для каждого дня lifetime */  
  
SELECT unique_users.lt,  
       ROUND (CAST (purchases.sum_purch AS float) /  
              CAST (unique_users.cnt_users AS float), 2) AS ARPU  
  
FROM unique_users  
  
LEFT JOIN (SELECT DISTINCT lt,  
                        SUM (amount) OVER (ORDER BY lt) AS sum_purch  
           FROM lifetime) AS purchases  
  
ON unique_users.lt = purchases.lt;
```

Посчитать ретеншн 1 и 3 дня для новых игроков, пришедших в мае. Сгруппировав их по недельным когортам.

Результат вывести в виде: неделя, число новых игроков, ретеншн 1го дня, ретеншн 3го дня.

```
WITH

/* находим номер недели для первого входа,
* дни лайфтайма для каждой сессии */

raw AS (SELECT s_o.user_id,
               s_o.time AS session_dt,
               first_exp.first_dt,
               strftime('%j', s_o.time) -
               strftime('%j', first_exp.first_dt) AS lt,
               first_exp.week

FROM session_open AS s_o

JOIN (SELECT user_id,
             time AS first_dt,
             strftime('%W', time) AS week

FROM session_open

WHERE session_index = 1
AND time BETWEEN '2022-05-01' AND
                '2022-05-31') AS first_exp

ON s_o.user_id = first_exp.user_id)

/* находим общее количество пользователей для каждой когорты,
* количество пользователей, вернувшихся в 1 и 3 дни
* лайфтайма для каждой когорты,
* рассчитываем retention rate для этих дней */

SELECT DISTINCT
    table_all.week,
    table_all.cnt_users,
    ROUND(CAST(table_1.lt_1 AS float) /
          CAST(table_all.cnt_users AS float) * 100) AS retantion_1,
    ROUND(CAST(table_3.lt_3 AS float) /
          CAST(table_all.cnt_users AS float) * 100) AS retantion_3

FROM (SELECT week,
             COUNT(DISTINCT user_id) AS cnt_users

FROM raw

GROUP BY week) AS table_all

LEFT JOIN (SELECT week,
                 COUNT(DISTINCT user_id) AS lt_3

FROM raw

WHERE lt = 3

GROUP BY week) AS table_3
```



```
LEFT JOIN (SELECT week,  
                COUNT(DISTINCT user_id) AS lt_1  
            FROM raw  
            WHERE lt = 1  
            GROUP BY week) AS table_1  
ORDER BY table_all.week;
```