



> Конспект > 7 урок > Дизайн А/В эксперимента

>Оглавление

>Оглавление

- > [Для чего нужны А/В тесты](#)
- > [Основы А/В тестирования](#)
- > [Метрики](#)
 - [Примеры метрик](#)
 - [Приоритетность метрик](#)
- > [Ошибки при принятии решений](#)
 - [Ложноположительная ошибка \(False Positive\)](#)
 - [Ложноотрицательная ошибка \(False Negative\)](#)
- > [Разбиение пользователей](#)
- > [Этапы А/В эксперимента](#)
 - [Формирование выборки](#)
 - [Оценка эффекта](#)
 - [Этапы проведения](#)

> Для чего нужны А/В тесты

- Разработка любого продукта — процесс постоянных улучшений
- Ожидаемое улучшение не всегда на самом деле улучшает продукт — новый функционал может быть никому не нужен

или реализация портит всю затею

- Каждая идея требует проверки — иначе наш продукт уйдёт не в ту сторону

Проверять идеи (гипотезы) можно по-разному:

- Обсудить с коллегами
- Опросить пользователей
- Пообщаться подробно с группой пользователей

Однако чаще всего этого недостаточно, а слова не всегда подтверждают фактическое поведение пользователей.

Опросы никак не помогут нам:

- Оценить новый алгоритм ранжирования фотографий в социальной сети
- Определить, не сломалось ли что-то при переезде инфраструктуры на новые технологии
- Проверить другой алгоритм ценообразования или новые инструкции саппортов

> Основы A/B тестирования

- Основная идея — тестируем 2 версии приложения / алгоритмов / ... на двух независимых группах пользователей
- Сравнение производим по значению метрик
- Ожидаем, что группы пользователей эквивалентны (из одной генеральной совокупности) — ведут себя одинаково (нулевая гипотеза), если изменений нет, метрики остаются теми же
- Наблюдения (пользователи) независимы
- Мы проверяем метрики на разных пользователях — метрики всегда будут различаться
- Статистические критерии помогают отличать шум от достоверных изменений

Любое A/B тестирование должно включать в себя:

- Метрики, по которым будем оценивать изменения
- Формирование групп (разбиение)
- Сбор метрик (как много наблюдений нужно собрать)

> Метрики

Заранее должны знать, что мы будем измерять.

Обычно в продукте уже есть устоявшийся набор метрик:

- Основные метрики — средний чек, конверсия в покупку, выручка на пользователя (характеризуют состояние бизнеса)
- Вспомогательные (прокси, опережающие) метрики — добавления товаров в корзину, конверсии этапов

воронки, оценки пользователей

- Также есть метрики, на которые мы влияем своим изменением (как часто люди находят наши товары при поиске)

В идеале мы бы измеряли прибыль компании, но это сложно отслеживать напрямую.

- Многие метрики не чувствительны к малым изменениям в продукте (например, средний чек)
- Чувствительные прокси-метрики позволяют оценивать качество изменений

Примеры метрик

Сервис для заказа такси

- Обновили дизайн карты для выбора точки
- Основные метрики — выручка на пользователя, конверсия из открытия приложения в поездку, количество поездок на пользователя в неделю
- Вспомогательные метрики — конверсии по этапам воронки, продолжительность сессии
- Количество нажатия для выбора точки, время на выбор точки — метрики, на которые влияем в первую очередь

Приложение для ведения заметок

- Обновили интерфейс форматирования текста

- Основные метрики — выручка на пользователя, количество заметок на пользователя в неделю, возвращаемость пользователя на следующий день
- Вспомогательные метрики — время создания заметки, количество открытий заметок в день
- Количество заметок с форматированием текста в день, доля использованного функционала для форматирования — метрики, на которые влияем в первую очередь

Приоритетность метрик

Когда метрик много, нужно знать, что важнее.

- Основные метрики можно условно разделить на денежные и продуктовые
- В каждом продукте могут быть свои главные метрики
- На стадии активного роста обычно важны продуктовые метрики
- На поздних стадиях важны денежные метрики
- Когда с одним показателем всё хорошо, то акцент может перейти на другой

> Ошибки при принятии решений

Мы используем статистический аппарат для проверки гипотезы, поэтому можем столкнуться с ложноположительными и ложноотрицательными ошибками.

Ложноположительная ошибка (False Positive)

- Сравнивали две группы, ошибочно посчитали изменение улучшением, выкатили новый алгоритм на всех пользователей
- Фиксируется за счёт выбора уровня значимости и корректности системы тестирования

Ложноотрицательная ошибка (False Negative)

- Сравнивали две группы, ошибочно не заметили улучшения, не выкатили новый алгоритм на всех пользователей

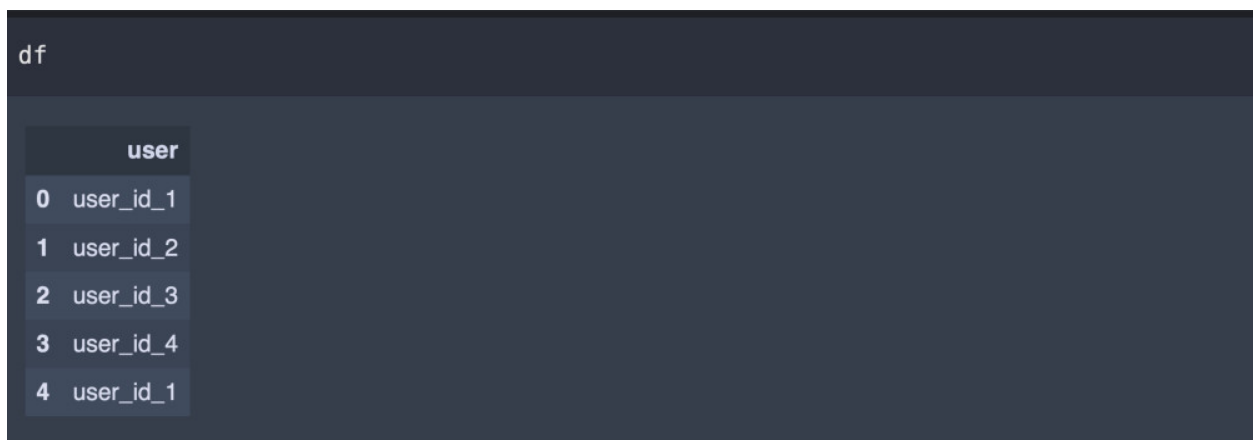
- Определяется мощностью статистических критериев и количеством данных

> Разбиение пользователей

Цель — сделать контрольную и тестовую выборку максимально похожими друг на друга.

- Достигается за счёт случайного разбиения пользователей
- Разбиение должно быть воспроизводимо (конкретный пользователь относится к определенной группе)
- Обычно реализуется через применение хэш-функции к идентификатору пользователя с некоторой солью, в зависимости от проводимого эксперимента

Представим, что у нас в выборке некоторое количество пользователей:



```
df
```

	user
0	user_id_1
1	user_id_2
2	user_id_3
3	user_id_4
4	user_id_1

Select an Image

Есть индикатор пользователя.

Соль (salt) — строчка, соответствующая конкретному эксперименту.

```
import hashlib

salt = 'my_first_experiment'
user = 'my_user_id'

value_str = user + salt
value_num = int(hashlib.md5(value_str.encode()).hexdigest(), 16)
value_num % 100
```

86

Select an Image

Вычисляем md5 hash от суммы строк id пользователя и salt, переводим в число берем остаток от деления на сто — это будет процент — группа, в которую попадает пользователь (>50% — вторая группа).

При повторном запуске функции получим тот же результат (воспроизводимость).

Удобно оформить код в виде функции и применять, когда нужно определить группу пользователя:

```
def get_group(user, salt, group_count):
    value_str = user + salt
    value_num = int(hashlib.md5(value_str.encode()).hexdigest(), 16)
    return value_num % group_count

df['group'] = df.user.apply(lambda user: get_group(user, salt, 4))
df
```

	user	group
0	user_id_1	0
1	user_id_2	1
2	user_id_3	3
3	user_id_4	1
4	user_id_1	0

Select an Image

> Этапы А/В эксперимента

Формирование выборки

Одно из требований — данные должны быть репрезентативны

- Ожидается, что собранная нами выборка хорошо описывает ГС
- Должны содержать всю неделю
- Не включать много праздников
- Содержать недельные циклы
- Выборка пользователей случайная
- Количество необходимых данных зависит от ожидаемого эффект и используемых критериев

Оценка эффекта

После сбора данных мы получаем две независимые выборки (используем статистические критерии для независимых выборок)

- Обычно применяем статистические критерии для проверки равенства двух средних
- Если отвергаем нулевую гипотезу, значит, считаем, что есть изменение в метрике
- Уровень значимости фиксируется заранее

Этапы проведения

1. Выбор метрик
2. Выбор размеров групп и продолжительности эксперимента
3. Запуск эксперимента и сбор метрик
4. Оценка результатов