3. Дизайн эксперимента и подготовка

Выбор метрик

Виды метрик

По-юзерные метрики

Сначала получаем значение метрики на пользователя, затем из этих значений получаем среднее.

Таким образом, числитель - сумма значений по пользователям, знаменатель - кол-во пользователей.

Например, таким образом считаются денежные метрики типа ARPU/ARPPU. В числитель мы берем сумму всех заработанных денег (GMV), в знаменатель мы берем количество пользователей. В результате получаем **среднее количество GMV на пользователя**.

Ratio метрики

Ratio метрики считаются несколько сложнее, чем поюзерные.

Разберем на примере метрики “средний чек” (AOV - average order value). Когда мы считаем средний чек, мы не суммируем и не усредняем значения на пользователя - в таком случае смысл метрики потеряется (т.к. нас интересует именно средний чек заказа). Проблема состоит в том, что в таком случае оценка может смещаться за счет того, что некоторые пользователи могут делать больше заказов, чем другие, соответственно “вес” таких пользователей в расчете среднего будет увеличиваться. Для того, чтобы учесть такое влияние используются разные статистические методы, мы используем линеаризацию (на этом подробнее останавливаться не будем, эта тема лежит за пределами этой инструкции).

Момент, который нужно учитывать - для Ratio метрик сложнее оценивать MDE и в целом проводить какие-либо оценки, поскольку мы применяем к ним дополнительную трансформацию.

Конверсия тоже может быть Ratio метрикой - например, когда мы считаем конверсию не на пользователя (сконвертировался ли он хотя бы раз за период эксперимента), а на каждый заход. Например, CTR - это ratio метрика.

Базовые, целевые и информационные метрики

Некоторые метрики имеют разную функцию внутри одного эксперимента.

Базовые - метрики, которые мы считаем важнейшими для всего продукта. Они расчитываются всегда для каждого теста. Сюда входят основные денежные метрики, gross profit и сквозные (из захода в заказ) конверсии.

Целевые - метрики, которые вы выбрали как критерий для принятия или отвержения вашей гипотезы. Таких метрик может быть не более 3-х. По этим метрикам мы считаем MDE и принимаем решение о длительности теста.

Информационные - метрики, которые чаще всего не влияют на принятие решения (или не являются основными критериями), но нужны для сбора дополнительной информации. Например, это могут быть метрики, которые позволят лучше понять поведение пользователей или получить дополнительные знания о природе полученного эффекта.

Выбор целевых метрик

При выборе целевых метрик лучше вернуться к вашей гипотезе и подумать, какие метрики помогут вам понять, подтверждается ли ваша гипотеза. Чем подробнее вы продумали вашу гипотезу (например, за счет чего будет расти метрика), тем проще вам будет выбрать целевые метрики.

Задача целевой метрики - по ее изменениям принять однозначное решение. Поэтому, важно:

- Заранее принять, какие метрики для вас целевые (плохо: провести эксперимент, а потом решать на основе каких метрик вы принимаете решение).
- Целевых метрик должно быть не более 3-х. Иначе - вы усложняете себе же принятие решения (допустим, выбрали 5 метрик, из них 3 выросли и 2 упали - начинаются сложности).

В идеале, хорошая целевая метрика должна быть так же и чувствительной. Понятно, что всегда хотелось бы поднимать средние чеки и ARPPU, но это не всегда возможно. Поэтому, если вы понимаете, что можете принять решение по более чувствительной узкой метрике (конверсия из этапа в этап, кол-во товаров, etc) или вы понимаете, что вашего эффекта недостаточно, чтобы прокрасить денежные метрики за адекватное время - покопайте в более низкоуровневые метрики. Это позволит вам сократить время на принятие решения, провести больше тестов и принести больше пользы бизнесу за счет **небольшого** снижения точности расчета эффекта и уверенности в результате.

Часто лучше провести 2-3 эксперимента с уверенностью 85%, чем один эксперимент с уверенностью 95% за то же время.

Размер трафика

Триггеринг

Не все пользователи **реально** участвуют в АВ-тесте. Например, если вы тестируете изменение на чекауте, а пользователь Вася зашел в приложение и только потыкал в каталог - он ваше изменение не увидит. Таким образом, ваше изменение никак не могло повлиять на Васю. Такие Васи могут быть во всех группах теста, они ничем друг от друга не отличаются, поэтому мы можем предполагать, что их метрики в разных группах не будут отличаться. Таким образом, мы имеем некоторое количество юзеров в каждой группе, на которых тестируемая фича никак не влияет.

Другой пример - у вас фича на **необязательном** этапе воронки. Например, вы добавили какую-нибудь плашку в личный кабинет. В личный кабинет заходят не все пользователи, и более того, возможно делать заказы и жить счастливую жизнь совсем не заходя в личный кабинет.

Длительность теста

Перед проведением эксперимента нужно оценить его необходимую длительность для заданного размера эффекта.

Ключевые концепции:

- Ожидаемый эффект - эффект на метрику, который, по вашей оценке принесет фича;
- MDE - минимальный детектируемый эффект для заданного количества наблюдений, зависит от среднего значения метрики и дисперсии. Если MDE у нас 1%, изменение в 0.9% мы прокрасить не сможем;
- Количество наблюдений (он же "трафик") - количество пользователей (заказов, магазинов, etc), которые побывают в эксперименте и по которым мы сможем посчитать метрики
- Длительность эксперимента - оценка срока проведения эксперимента на основании необходимого количества наблюдений

Для оценки длительности эксперимента вам нужно:

- Взять ваши целевые метрики
- Для каждой из этих метрик оценить ожидаемый эффект. Как это сделать - есть много способов. Можно оценить прирост п ри помощи аналитика, взять бенчмарки с рынка, посмотреть приросты метрик по аналогичным фичам и т.д.
- При помощи аналитика рассчитать необходимое количество трафика при заданном MDE (MDE должен быть меньше или равен вашему ожидаемому эффекту, если он выше - такое изменение не прокрасится). Дашборд калькулятор MDE
- На основе необходимого количества пользователей рассчитать необходимую длительность (с учетом триггеринга)

Количество групп

В идеальном случае, мы стремимся к тому, чтобы в АВ-тесте было две группы - контрольная и тестовая. Однако, иногда хочется проверить сразу несколько вариантов.

Это нормально, но стоит помнить, что в таком случае придется применять поправки и занижать наш требуемый уровень p-value. Это приведет к тому, что вам придется проводить эксперимент дольше... Подробнее: https://habr.com/ru/companies/yandex/articles/476826/

Юнит сплитования

Пересечения

Существует всего три способа проведения нескольких экспериментов.

Разведение по времени

Самый простой способ. Мы проводим Тест 1, а затем проводим Тест 2 после него. Все просто.

Достоинства:

- **Максимально "чистые" эффекты.** Никаких взаимовлияний.
- **Тест 2 уже учитывает контекст Теста 1.** Т.е. Тест 1 либо выкатили, либо нет. Фичи тестируются в максимально "продовой" среде.

Недостатки:

- **Тесты проводятся миллион лет.** Если мы можем проводить один эксперимент за раз, то это неизбежно приведет к огромным очередям.

Разведение по трафику

Способ, когда мы делим трафик между несколькими тестами. Например, мы отдаем 50% трафика Тесту 1 и 50% трафика Тесту 2. Таким образом, 25% трафика уходит контрольной группе Теста 1, 25% уходит тестовой группе Теста 1, и аналогично для Теста 2.

Это делается при помощи сихронизации соли между тестами.

Соль - это такой random seed для теста, чтобы синхронизировать "случайность". Если у обоих тестов одинаковая соль, то если пользователь Вася попал в бакет 139 в Тесте 1, то он попадет в бакет 139 и в Тесте 2. Таким образом, можно задать бакеты 0-500 для Теста 1 и бакеты 500-1000 для Теста 2, и в Тесте 2 Вася совсем не поучаствует.

Достоинства:

- **Тесты не аффектят друг друга.** Совсем. Никаких взаимовлияний.

Недостатки:

- **Фичи тестируются в "нерепрезентативной" среде.** Если мы изолируем наши фичи друг от друга и получаем некоторый эффект, то при раскатке этих нескольких фичей мы рискуем получить неожиданный исход, т.к. мы не знаем как эти две фичи живут вместе.
- **Соль имеет свойство "протухать".** Есть тенденция к тому, что разведения тестов становятся длительными. Как это работает? У нас есть некая соль X. Мы завели два эксперимента, и развели их по трафику. Тест 1 закончился раньше, и мы на этот трафик завели новый эксперимент. Затем закончился Тест 2, и на эти 50% трафика мы завели еще два эксперимента по 25% трафика. И так может продолжаться бесконечно, что приводит к тому что соль не "обновляется" совсем. Это может привести к тому, что бакеты 0-500 и 500-1000 через некоторое время разойдутся и перестанут быть однородными, т.к. они получали неодинаковый опыт.

Пересечение

Мы просто запускаем два эксперимента одновременно и не синхронизируем их по соли. Тогда эти эксперименты посплитуются "перпендикулярно". Что это значит? Что контрольная группа Теста 1 равномерно попадет в контрольную и тестовую группы Теста 2, и т.д.

Таким образом, тесты не должны смещать эффекты друг друга.

Достоинства:

- **Мы можем проводить очень много тестов за раз.** Никаким другим способом постоянный поток тестирования множества гипотез не обеспечить.
- **Мы можем узнавать о взаимовлияниях фичей.** Мы не знаем, в каком состоянии (т.е. с какими фичами) будет продукт тогда, когда мы выкатим тест. Поэтому тестировать такие вещи вместе может быть полезно.

Недостатки:

- **Иногда мы можем обнаружить, что некоторые фичи имеют неаддитивный эффект.** Это когда две хорошие фичи при их сочетании превращаются в одну плохую. Однако, такие случаи встречаются не так часто. Классический пример - в одном из тестов цвет кнопки изменяется на синий, а в другом - текст кнопки изменяется на синий. В результате часть людей видит синюю кнопку с синим текстом.
- **При небольшом количестве трафика иногда тесты могут пересечься неравномерно.** Решение для этого - брать больше трафика.

Вывод

Тесты лучше пересекать всегда, когда это возможно. Разводить тесты по трафику стоит только если у вас есть серьезное основание для этого - например, если вы заранее понимаете, что фичи “не живут” вместе.