

Тестовое задание

SQL

Задание: написать SQL-запросы оптимальным способом. Предпочтительно использовать ClickHouse SQL диалект.

Необходимо рассчитать:

1. Заплаченную сумму со 2-го по 7-й (включительно) платеж на игрока по всем данным. Результат: 1 число суммы платежей
2. Среднее количество логинов и платежей в первые 4 недели с момента установки игры. Результат: 2 числа - среднего по логинам и платежам
3. На каждый день жизни игрока подсчитать сумму его платежей в следующие 7/30 дней. Показать среднее значение этой суммы по всем игрокам в течение первых 10 дней жизни игрока. Результат: 10 строк по каждому дню жизни игрока и 3 колонки (день жизни, средняя сумма платежей в следующие 7 дней, средняя сумма платежей в следующие 30 дней)

Формат ответа: SQL-запросы и полученный результат.

Данные для запроса

Для расчёта использовать данные из файла **task1_facts.txt**, где содержится готовая для использования схема таблицы, например, на сервисе <https://www.db-fiddle.com>

Описание колонок:

- user_id - уникальный id игрока
- event_name - название действий
 - 'install' - установка игры. Если у игрока отсутствует install, то полагаем, что когортные расчёты произвести не получится
 - 'login' - заход в игру
 - 'payment' - совершение платежа
- event_time - время совершения действия
- currency - валюта платежа (актуально для event_name = payment)
- amount - сумма платежа (актуально для event_name = payment)

A/B test

1. Частотная статистика

Проводится а/б тест стартовых предложений в магазине мобильной игры.

Две группы:

- 1 - Контрольная
- 2 - В 5 раз снижена цена и в 2 раза наполнение стартового набора - предложения, которое наиболее часто покупают в качестве первого платежа.

По данным в **csv** файле оцените эффект нововведения в тестовой группе и сформулируйте выводы.

Метрики для расчёта теста определите самостоятельно исходя из доступного набора данных (в набор оцениваемых метрик должен входить ROAS - return on ads spend).

Данные для расчёта теста в файле **task2_ab_dataset.csv**.

Описание колонок:

- user_id - уникальный id игрока
- ab_group - тестовая группа
- install_time - время установки игры
- payments_time - время совершения платежа
- amount - стоимость купленного пакета
- cost - стоимость привлечения игрока

Формат ответа: jupyter notebook с расчётами и выводами.

2. Байесовская статистика

Реализуйте модель с помощью PyMC которая сравнивает конверсию в плательщика между двумя группами.

Требования:

- Критерий остановки - 0 не входит в 95% HDI
- Сделайте необходимые допущения по недостающей информации относительно сетапа теста (например, по объёму выборки, априорному распределению, остановке сбора данных и т.д.)
- Как пример работы модели, используйте данные из прошлого задания для визуализации результата
- Кратко прокомментируйте выбор параметров описывающих результат

Формат ответа: jupyter notebook с реализованной моделью и комментариями.