# Этапы работы над дипломным проектом.

Дисклеймер: я делаю всё только в sql и python, ни одной экселевской таблички нет.

Этап 1: получение данных.

Данные хранятся в таблицах Metabase, получаем данные за апрель 2021 года с помощью sql-запроса, соединяя две таблицы: orders и users. Обратите внимание, что для разных таблиц используются разные ключи. Таблица region не используются, данные там ничтожны.

Получившаяся в результате запроса таблица выгружается в csv-файл.

Этап 2: анализ данных.

Анализ данных проводится в python. Для этого читается файл из п. 1. При парсинге учитываем, что часть столбцов – это даты, поэтому сразу парсим их как даты, с указанием ключа dayfirst=True.

Смотрим размер полученного датафрейма и статистику. Видим, что выбросов нет.

Создаём и наполняем столбцы в фиксированными затратами, валовыми затратами, и долей валовых затрат в размере транзакции.

В итоге получаем рабочий датафрейм со всеми необходимыми для расчётов данными.

Этап 3: расчёт базовых показателей.

|  |
| --- |
| Поле **Purchases** - количество строк в выгрузке**,** то есть количество покупок за апрель  Поле **Unique Users** - количество уникальных пользователей  **Purchase per User** - это среднее количество покупок на одного юзера.  **Total Comission -** сумма по полю commission  **Total Value** - сумма по полю transaction\_value  **Margin** - отношение **Total Comission** к **Total Value**  **Fixed Costs**  **processing\_cost** - сумма по полю processing\_cost  **integration\_cost** - сумма по полю integration\_cost  **Marketing Costs**  **promocode\_cost** - сумма по полю promocode\_cost  **Costs** = **Fixed Costs + Marketing Costs**  **Costs share** - отношение **Costs** к **Total Value** |

Этап 4: фиксация параметров модели.

Определяем и присваиваем следующие параметры:

|  |
| --- |
| **Promo\_costs** = 50000 (билет в Дубай будет выдаваться одному случайному клиенту вне зависимости от общего количества клиентов)  **Purchase per User =** 10 (наши *ожидания* относительно среднего кол-ва покупок на одного клиента)  **Loyalty Cost Share** = 1% (1% от суммы транзакции будет возвращен клиенту в рамках программы Loyalty)  **Average Transaction Value ATV** = 3000 (наши *ожидания* относительно среднего чека при покупке) |

Этап 5: моделирование.

Создаём функцию, которая на вход берёт 4 параметра (из п. 4, со значениями по умолчанию), и проводим расчёт следующих показателей:

|  |
| --- |
| Purchase per User \* ATV \* Users, где Users от 1 до n (как раз такие нужно найти n, кол-во человек, при котором вся затея окупится)  Если это значение умножить на Margin, то получим общую прогнозируемую выручку.  Затраты на программу Loyalty;  Выручка за вычетом программы Loyalty;  Выручка за вычетом программы Loyalty и затрат на приз;  Валовое значение костов;  Операционная прибыль, то есть выручка минус операционные косты, без учета костов на Loyalty и приза;  Чистая прибыль. |

На выход функции отдаём 2 списка: с количеством пользователей и с чистой прибылью.

Этап 6: визуализация.

Берём два списка из п. 5, и с помощью метода plot из библиотеки matplotlib строим график выхода программы лояльности на безубыточность.

Этап 7 (необязательный): презентация.