№56 FINAL PROJECT
Выполнен студентом группы SDA-170 Голомбовским Николаем

Что находится в данном файле?

Для окончания обучения по курсу "Аналитик данных" в онлайн-школе Skillfactory необходимо решить финальное задание №56. Решение ожидается с помощью SQL (запрос в базу данных на получение исходных таблиц) и Python (обработка, очистка и визуализация данных).

Так же как и все другие выпускники Skillfactory, я выполнил задание №56 и получил за него максимальный балл.

Однако, за время обучения в Skillfactory я стал фанатом Power BI. Я был несколько растроен тем, что для решения задания №56 не требуются знаний и навыков Power BI. Поэтому я решил исправить данную "несправедливость" и решить данное задание №56 исключительно средствами Power BI.

Как оказалось, до меня такая идея не приходила в голову ни одному студенту. Как я об этом узнал? Во-первых, ментор финального задания послал меня к ментору блока по Power BI, аргументируя тем, что не достаточно знаком с данной системой. Во-вторых, ментор Power BI упомянула, что неоднократно принимала участие в приёмке финальных заданий у студентов Skillfactory, но при этом ни разу не упомянула, что кто-то ещё пытался решить задание №56 с помощью Power BI.

Возникает вопрос - а зачем я обращался к этим 2 менторам Skillfactory? Дело в том, что решение далось мне не сразу и я обращался к ним за советами. Скажу сразу, ментор финального задания, как я уже указал, был не достаточно компетентен для оказания помощи. Ментор Power BI пообещала помощь, но пока она "раскачивалась", я уже закончил решение этого задания с помощью Power BI.

Исходные данные

На вход задания № 56 поступает 2 таблицы - данные о покупках студентами некоторой онлайн-школы:

Таблица "carts" — данные о пользовательских корзинах

Promo Code ID — ID промокода, если он есть

Purchased At — дата оплаты

User ID — ID пользователя

Created At — дата создания корзины

Updated At — дата последнего обновления информации

ID — идентификатор корзины

State — состояние оплаты

Таблица "cart items" — данные о курсах, которые пользователи добавили в корзину

Created At — дата создания события

Resource Type — тип продукта

Resource ID — ID курса

Cart ID — идентификатор корзины

Updated At — дата последнего обновления информации

ID — идентификатор операции

Чуть позже я дам некоторые пояснения по выполнения данного задания и по содержимому ряда таблиц. Дело в том, что решения задания №56 не далось мне сразу, а потребовало некоторого времени. В связи с этим некоторые блоки задания имеют несколько вариантов решения. Если вы самостоятельно откроете закладку с таблицами, то у вас могут возникнуть вопросы относительно необходимости иметь таблицы с примерно одинаковым названием и идентичным наполнением. Я поясню почему и какие таблицы за что отвечают.

№56 FINAL PROJECT
Выполнен студентом группы SDA-170 Голомбовским Николаем

ЛЕГЕНДА

Итак, вы работаете аналитиком в онлайн-школе MasterMind.
Уже в конце рабочего дня вам пишет расстроенный продакт-менеджер. Несчастный Григорий крайне устал от того, что новые курсы, созданные с той же любовью, что и прежние, не пользуются особой популярностью среди пользователей — несмотря на все усилия отдела маркетинга.

ЦЕЛЬ

Подготовить основу рекомендательной системы.

ЗАДАЧИ

Итак, продакт ожидает получить от вас рекомендательную систему, благодаря которой можно будет предлагать клиентам интересные им курсы и тем самым повышать средний чек.
Вы решаете, что изначальным воплощением этой системы может стать таблица, в которой курсам будет соответствовать по две рекомендации.
Кроме того, вы планируете вместе с отчётом (таблицей рекомендаций) скинуть продакту ещё и все написанные в процессе скрипты, чтобы было меньше вопросов по решению :) Ну, и раз в код будут смотреть не только ваши глаза, вы считаете необходимым снабдить его комментариями, которые бы разъясняли, что где и почему вы делаете.
Также вы понимаете, что перед внедрением фичи коллеги решат провести A/B-тест и вас скорее всего привлекут к анализу результатов.
Перспективы ясны, можно переходить к формализации задач.

КОНКРЕТНЫЕ ШАГИ (ФОРМАЛИЗОВАННЫЕ ЗАДАЧИ)

Обдумав план предстоящей работы, вы понимаете, что действовать нужно по привычной схеме:
1. Познакомиться с датасетом, подготовить и проанализировать данные с помощью SQL.
2. Обработать данные средствами Python.
3. Составить итоговую таблицу с рекомендациями, снабдив её необходимыми комментариями, и представить отчёт продакт-менеджеру.
4. Проанализировать результаты A/B-теста, проведённого после внедрения фичи, и сделать вывод.

РЕЗУЛЬТАТ ПРОЕКТА

Итогом работы станет файл, содержащий результаты всех промежуточных этапов: скрипты с комментариями, таблица рекомендаций и выводы.

№56 FINAL PROJECT
Выполнен студентом группы SDA-170 Голомбовским Николаем

Как я уже писал ранее, некоторые части задания №56 мне не удалось решить сразу с помощью только Power BI. Здесь я привожу описание нескольких вариантов решений одних и тех же частей и в каких таблицах находится содержимое их выполнения. В конечном итоге ВСЕ части задания №56 были мной решены исключительно с помощью Power BI.

Трансформация таблицы "студент-курс" в таблицу пар "курс-курс"

У нас имеется таблица "студент-курс". В ней содержатся все купленные учебные курсы данными студентами, причём каждый студент купил не менее 2 курсов (почему именно 2 будет понятно чуть ниже). В рамках этапа задания нужно преобразовать данную таблицу в таблицу пар "курс-курс", в которой содержатся всевозможные пары курсов, покупаемые студентами.
Поясню. Например, некоторый студент №1 купил курсы №1, №2 и №3. В таблице "студент-курс" у нас имеются записи (Ст1, Ку1), (Ст1, Ку2) и (Ст1, Ку3). Нам нужно преобразовать эти строки в следующие записи в таблице "курс-курс" - (Ку1, Ку2), (Ку1, Ку3) и (Ку2, Ку3). Другими словами на выходе мы получаем всевозможные 2-элементные комбинации курсов, купленных студентом №1.
Комбинаторика нам даёт оценку - если студент купил "n" курсов, то количество разных комбинаций из 2 курсов равно "n * (n-1) / 2".
1. Вначале данную задачу я решил с помощью Python вне Power BI и уже готовый результат внёс как источник с расширением "csv" в Power Query. В результате имеем таблицу "corses_doubles".
2. Потом я разобрался как можно интегрировать код Python (тот самый, который решает данную задачу) в Power Query - получил таблицу "courses_doubles_Python".
3. Далее я искал функцию языка DAX, которая мне могла бы помочь решить данную задачу. И я нашёл её - получил таблицу "cources_doubles_DAXAll".

Но на этом не всё закончилось...
Когда я попытался преобразовать таблицу "студент-курс" ("merge_carts_good") целиком, то получил от Power BI сообщение о нехватке памяти для решения этой задачи. При этом на моём компьютере установлено 16 гигабайт ОЗУ, количество дисковой памяти исчисляется сотнями гигабайтов.
В связи с этим я решил данную таблицу разделить на 2 - "merge_carts_good2" в которой содержатся покупки студентов, купившие ровно 2 курса, и "merge_carts_good3_" в которой содержатся покупки студентов, купившие 3 и более курсов.
После их преобразования я получил соответствующие таблицы "cources_doubles_DAX2" и "cources_doubles_DAX3_", которые в итоге я объединил в одну "cources_doubles_DAXAll".

Поиск рекомендаций для каждого курса - покупка каких ещё 2 курсов будет иметь наибольший успех у студентов

Поиск данный рекомендаций простой - для каждого курса мы подбираем ещё два, с которыми первоначальный курс уже покупался чаще всего. Именно для этого нам и нужна построенная таблица пар курсов "курс-курс".
Первоначальный вариант решения был следующим. Я построил таблицу из одного столбца, в который записал все имеющиеся учебные курсы (все кроме одного. который так и не был ни разу куплен ни одним студентом). Затем последовательно к данной таблице я добавлял по одному столбцу, в которых рассчитывал промежуточные данные и в конце рассчитывал два рекомендуемых к покупке учебных курса. Таким образом получилась таблица "Courses".
Затем мне пришла идея создать точно такую же таблицу, но одним скриптом на языке DAX - все столбцы одним махом. Так появилась полностью аналогичная таблица "Courses2".

Надеюсь, данных пояснений будет вам достаточно для лучшего понимания списка таблиц и их содержимого.
Так же, внутри кода я по возможности прописывал комментарии с пояснения к коду DAX. Надеюсь, этого так же окажется вам достаточно для понимания алгоритмов кода DAX.

Соответствие между состоянием корзины и датой оплаты за неё по ВСЕМ студентам

state	2017	2018	**Total**
created	17 641		**17 641**
pending	5 352		**5 352**
successful		26 043	27 605	**49 006**
Total	**22 940**	**26 043**	**27 605**	**68 989**

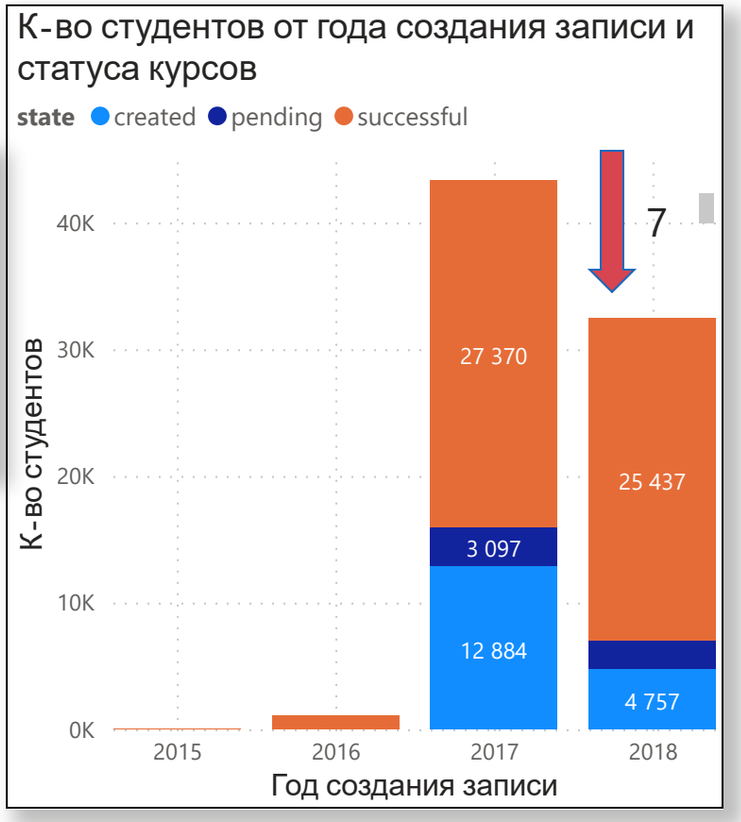
1. В имеющейся базе данных имеется ПРЯМАЯ связь между годом ПОКУПКИ корзины и её СОСТОЯНИЕМ - ВСЕ выкупленные корзины имеют состояние "successful".

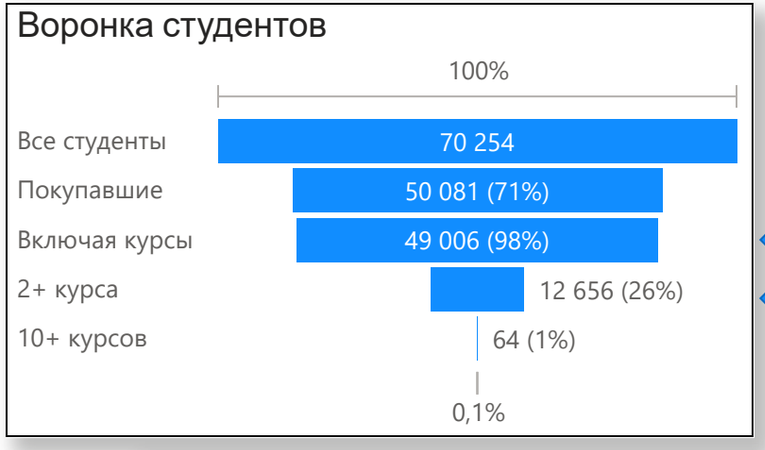
2. Следуют отметить, что покупки проводились в 2017 и 2018 годах - это является ответом на **Задание 56.3.1** "**Продажи за какие годы есть в ваших данных?**"

Данные по студентам и годам

1. 2/3 (68%) корзин студентов были оплачены имеют состояние "successful".

2. Количество студентов, СОЗДАВШИХ корзины в 2018 году, на 7% меньше по сравнению с 2017 годом.





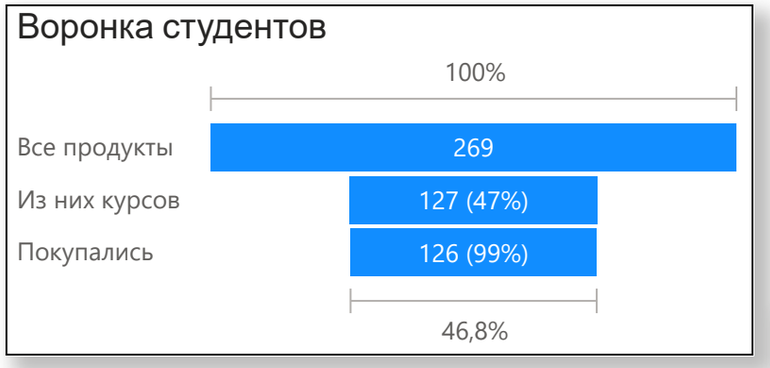
Среднее количество курсов на 1 студента

1,44

1. **Задание 56.3.2 "Сколько клиентов покупали курсы?"** - 49 006 студентов.

2. **Задание 56.3.5 "Сколько клиентов купили больше одного курса?"** - 12 656 студентов. Это примерно 1/4 от всех студентов, купивших хотя бы один курс. Или 3/4 от купивших курс купили ТОЛЬКО один учебный курс.

3. **Задание 56.3.4 "Каково среднее число купленных курсов на одного клиента?"** - 1.44 курса на одного студента, купившего хотя бы один курс.



Данные об учебных курсах и их парах

Всего различных курсов

127

1. Среди списка продуктов, продаваемых нашей онлайн-школой, 47% приходится на учебные курсы. При этом на те же самые учебные курсы приходится уже 97% от всех продаж нашей школы.

2. За 2017 и 2018 годы было продано 127 различных курсов - это является ответом на **Задание 56.3.3** "**Сколько всего различных курсов?**"

Популярность пар курсов

unique_cources	Count of course1
551, 566	797
515, 551	417
489, 551	311
523, 551	304
566, 794	290
489, 515	286
490, 566	253
490, 551	247
570, 752	247
569, 572	216
515, 523	213
553, 745	212
489, 523	206
569, 840	204
Sum by column	**40017**

Количество различных пар курсов

3989

1. **Задание 56.3.6 "Сколько различных пар курсов встречаются вместе в покупках клиентов?"** - 3 989.

2. **Задание 56.3.7 "Найдите самую покупаемую пару курсов? Какие у них ID?"** - лидером является пара курсов 551 и 566. Она встречается 797 раз

P.S. На данный момент таблица ПАР покупаемых курсов импортирована в Power BI, будучи до этого экспортирована из Python. Другими словами, не удалось перевести таблицу покупаемых студентами курсов в таблицу покупаемых пар одними и теми же студентами.
К сожалению ПОКА не удалось реализовать итоговую курсовую работа ИСКЛЮЧИТЕЛЬНО средствами Power BI - для этого ПОКА не хватает знаний. Но направление для исправления найдено - это использование кода Python в Power Query. Но это не реализовано в данной версии итоговой курсовой работы. Все ОСТАЛЬНЫЕ расчёты проведены ИСКЛЮЧИТЕЛЬНО возможностями и средствами Power BI.

Рекомендуемые для последующей покупки учебные курсы

Учебный курс	Суммарный рейтинг	Рекомендация №1	Рекомендация №2
551	1	566	515
566	2	551	794
515	3	551	489
490	4	566	551
514	5	551	515
489	6	551	515
523	9	551	515
745	9	553	516
794	9	566	551
570	10	752	507
502	11	551	566
840	12	569	572
552	13	551	523
571	15	1125	357
809	15	490	570
507	16	570	752
504	18	572	569
572	18	569	504
564	20	523	551
764	20	566	551
752	21	570	507
569	22	572	840
519	23	551	523
679	24	551	489
516	26	745	553
1125	26	571	912
357	28	571	356
1103	28	551	566
749	29	551	515

В таблице приводится результат САМОСТОЯТЕЛЬНОЙ работы в рамках итоговой курсовой работы по позиции "Аналитик данных" в Skillfactory.

В данной таблице приведены следующие колонки:

1. Номер учебного курса.
Здесь приведены ВСЕ продававшиеся учебные курсы в рамках анализа.

2. Суммарный рейтинг каждого курса на основе двух рейтингов:
- Количество студентов, которые покупали данный курс,
- Количество покупаемых ПАР курсов, в которых участвует заданный учебный курс.

В требовании задания данного столбца нет. Я его добавил, так как он используется при расчёте рекомендаций для курсов, которые редко участвует в парах покупаемых курсов. Так же, данный рейтинг можно использовать для выбора ДВУХ рекомендуемых курсов для конкретного студента - алгоритм рекомендаций может привести к тому, что какому то студенту будет рекомендоваться пул из более, чем 2 курсов для последующей покупки. Вот и предлагается выбирать из данного пула ДВЕ рекомендации, ориентируясь на данный сводный рейтинг.

3. Первый рекомендуемый для последующей покупки курс.
Он выбирается по следующему алгоритму:
- Выбирается НАИБОЛЕЕ ПОКУПАЕМАЯ пара курсов, в которой участвует данный курс. В качестве рекомендации для последующей покупки выбирается "парный" курс.
- Если количество покупок данной пары меньше 5, то происходит замена рекомендуемого курса. Выбирается СЛУЧАЙНЫМ образом курс из 20 НАИБОЛЕЕ покупаемых ПАР курсов. Уж лучше такой курс, чем редкопокупаемый.

4. Второй рекомендуемый для последующей покупки курс.
Алгоритм тот же, что и для рекомендации №1, только выбирается вторая по покупаемости пара курсов. Так же происходит замена рекомендуемого курса, если пара покупалась менее 5 раз.

ВНИМАНИЕ!
Конкретные значения рекомендуемых курсов могут отличаться от тех, которые были рассчитаны в Python. Это связано с тем, что для одного курса может существовать НЕСКОЛЬКО пар с одинаковой покупаемостью данных пар. В этом случае вполне возможно, что алгоритмы Python и Power BI использовали в расчётах разные пары, что привело к отличающимся рекомендациям курсов. Однако в этом случае, в рамках используемого алгоритма