



{ balkondev }



Пётр
Васильев



Максим
Лаврушин



Евгений
Блинов

Кейс МТС

Умные покупки



Сервис по увеличению доли банка в кошельке клиента

Автоматически рекомендует корзину покупок и зоны интересов персонально для клиента на основании истории как самого клиента, так и близких ему по покупательскому поведению, непрерывно обучаясь откликах клиента на сделанные офферы

{ Почему это срабатывает? }

Люди любят не забывать, но такое частенько случается...

Мы проявляем заботу и клиенте и напоминаем ему в удобный момент о важной для него покупке. Персональное напоминание поданное в идеальный момент - что может быть лучше?

Радость новых открытий

Предложим новый интересный кейс, которого у клиента еще не было, но был у наиболее похожих на него клиентов из той же группы, ранжированной по доходу и региону.

Что если чуть-чуть увеличить частоту покупок клиента?

1 новая покупка в месяц из наиболее частых категорий у 5% клиентов дает рост объема транзакций на 12%

Что может быть лучше новых открытий?

1 новая покупка в месяц из наиболее выстрелованных категорий у наиболее близких к нашему клиенту других наших клиентов, на 5% клиентов дает рост объема транзакций на 12%

{ Use cases }

Сценарий 1

Выдаем ТСП на основании регулярности закупок.

Фильтруем выданный ТСП по координатам(наиболее близкие к клиенту). Фильтруем ТСП по стоимости и предлагаем более дорогую группу с учетом скидки(смотрим скидку или кэшбэк).

Сценарий 2

Смотрим, сколько пользователь тратит в среднем в неделю. Предлагаем клиенту ТСП из выбранных похожими клиентами по поведению(фильтруем юзеров по группам дохода). Сравниваем траты последней недели со средним, и если пользователь потратил меньше - предлагаем ему большую скидку(смотрим скидку или кэшбэк).

{ Как делаем }

Выделяем номер дня и месяца.

Обучаем классификатор для предсказания релевантной корзины на выделенных параметрах `mcc_code` и юзер.

Получаем многоклассовый ответ.

Решаем задачу многоклассовой классификации - определяем зависимость на основании истории закупок.

Применяем методы фильтрации - ищем ближайшие ТСП по меньшему расстоянию между координатами.

Группируем клиентов по ТСП.

Методом (Пирсона, Жаккарда) определения сходства ищем похожих между собой клиентов.

Определяем разность в выборе ТСП активного юзера и массива похожих.

Выделяем разность как предложение.

Фильтруем ТСП по категории юзера и *ТСП*.

{ Дополнительные данные }

1. Ввели группы юзеров по уровню дохода - можно
2. Ввели координаты юзера и ТСП - можно использовать realtime координаты.
3. Временные сдвиги и скользящее среднее с окном в 3 дня
4. Ввели признак онлайн/оффлайн покупка - ранжировать ТСП по типу.

{ Как улучшить? }

Мы бы могли улучшить решение в дальнейшем добавив:

- Автоматическую чистку и нормализацию данных, обработку пропусков и выбросов
- Выявление и реагирование на аномалии (инцидент менеджмента)
- Кластеризация клиентов и ТСП в автоматическом режиме
- Расширение параметров совместной фильтрации
- Подбор наиболее оптимального классификатора базовой модели определения релевантной “корзины покупок” для клиента
- Геолокацию (реалтайм отслеживание позиции клиента) и Тип покупки (онлайн/оффлайн)
- Подбор оптимальных офлайн метрик определения точности модели
- Динамическое обучение “на лету”
- Онлайн мониторинг работоспособности модели на основе A/B тестов и обратной связи
- Реализация конвейера в функционале автономного сервиса

{ balkondev }

