**ML-фильтрация.**

1. Сначала нужно вывести пользователю первичные фильтры (5+-2). Для этого проводим кластеризацию сета на N1-кластеров. Алгоритм подбираем так, чтобы получить максимально различные кластеры (по тому же силуэту или другой метрике).

Количество кластеров N подбираем исходя из максимального кол-ва шагов клиента до результата. Чем больше N, тем меньше шагов потребуется.

1. Проводим оценку значимости признаков для каждого кластера. Приходит на ум – обучить модель классификатора sklearn и вывести вес фичей (признаков), предварительно отфильтровав признаки с сильной корреляцией, слабо-вариативные, с большинством отсутствующих значений. В данном случае каждый класс это кластер. Вес фичи – коэффициент при определённом признаке. Чем он больше (вес), тем в большей степени данная фича влияет на отнесение объекта (строки) к тому или иному классу.
2. Отбираем 5+-2 фичи с бОльшими весами. Это будут первичные фильтры.   
   Тут нужно оговориться. Если не использовать семантику и логику (гибридный подход), то можем получить в виде фичей какую-то дичь, о которой выбирающий кабель инженер-электрик (вроде меня - )) может даже не знать.
3. Первичные фильтры заполняются, возвращаются на сервер в модель классификации. Модель предсказывает (с некоторой вероятностью) кластер, к которому относится желаемый покупателем объект.
4. Предсказанный кластер делится на N2-кластеров. Запускается вторая итерация. Оцениваем вес признаков моделью классификации, отбираем кол-во фильтров в зависимости от их веса. Не обязательно 5+-2. Их может быть меньше, если остальные признаки незначительны. Признаки берём отличные от взятых ранее, само собой. Можно вообще убрать предыдущие признаки при обучении модели классификации – немного оптимизируем/ускорим процесс.
5. Повторяем до результата или его отсутствия.

|  |  |
| --- | --- |
| Преимущества | Недостатки |
| 1. Не акцентируем внимание на объектах. Только на данных 2. Можем задавать N-кол-во кластеров, исходя из того, насколько быстро нам нужно получить результат. Больше N – меньше кол-во итераций для достижения K-значения результирующей выборки | 1. Без добавления семантических проверок в фильтрах может появиться всякое 2. Чем больше N и чем меньше кол-во отбираемых признаков-фильтров, тем вероятность ошибки выше 3. Быстродействие, по сравнению с теми же семантическими запросами к БД, оставит желать лучшего |

Риски в целом:

Модель машинного обучения – алгоритм, на выходе которого мы получаем вероятность. То есть, чем меньше мы отбираем признаков (фильтров), тем вероятнее результатом вычислений станет угадывание. И вероятнее в результатах подборки появится то, что клиенту не нужно. Это палка о двух концах, где можно (теоретически) установить балансную точку между количеством фильтров до результата и вероятностью ошибки.  
Также остаётся проблема редких признаков каких-либо уникальных объектов, которые пользователь мог бы найти в 1 клик по нужному фильтру. Алгоритм выше затруднит его поиск. Или сделает невозможным.

Возможные варианты решения:

* Применение гибридного подхода (семантические проверки, логические проверки). Реализация усложнится. Недостатки нивелируются, преимущества, от части, тоже. Риск ошибиться станет ниже, но нужно будет осмысливать данные, акцентироваться на признаках.
* Выделение такого способа фильтрации отдельной вкладкой в качестве альтернативного. В большинстве случаев он ускорит поиск.

В случае если максимальное число признаков достигнуто ( 20+-), но размер предсказанного кластера всё равно достаточно большой, то возможно:

* Взять уникальный признак для каждого объекта (подсчёт COUNT DISTINCT значений по всем колонкам с выбором максимального числа) в последнем предсказанном кластере и по его значению провести финальную фильтрацию
* Имея 20+ признаков за все итерации и 5+-2 признака на последнем кластере – найти наиболее подходящие товары по всей БД на 20 признаках, либо на последнем кластере по 5+-2 признаках. Либо поместить в итоговый результат совокупность этих результатов. Сделать это можно SQL-запросом, либо посредством обучения модели-классификатора по всем позициям в БД/кластере на соответствующих признаках. В итоге получим результат на основе выбранных пользователем значений признаков.