**Рекомендательный ИИ для интернет-магазина**

Проблематика, кейс: необходима разработка интеллектуальной рекомендательной системы для интернет-магазина, которая позволит увеличить продажи

**Описание задачи:**

Исходя из данных о поведении посетителей на сайте интернет-магазина, необходимо разработать модель для рекомендации товаров пользователям. Модель должна предсказывать топ 25 товаров, которые пользователь купит с наибольшей вероятностью в следующие 3 дня (период тестового датасета). Метрикой качества является Recall по купленным товарам.

Например, в сформированном вашей моделью списке из 25 товаров, оказалось 3 из 4х, которые пользователь реально купил. Recall для этого пользователя составит ¾ = 0.75.

Для пользователей из тестовой выборки, которым не нашлось соответствия в вашем прогнозе - Recall составит 0. Пример приведен на рисунке, где в поле products\_y (прогнозные товары вашей модели) стоит значение NaN

При этом, топ товаров составляется ровно для 3000 пользователей, которые с наибольшей вероятностью совершат покупку. Т.е. ваш сабмит-файл не должен превышать 3000 объектов (пользователей) и должен соответствовать формату выгрузки, представленному в файле baseline.ipynb. В противном случает Recall посчитается, как ноль

В распоряжении имеется 4 набора данных, содержащий следующие поля:

stokman\_catalog\_preprocessed.pq

* add\_date - дата добавления товара в магазин
* shop\_id - идентификатор магазина
* product\_id - идентификатор товара
* category\_id - идентификатор категории товаров
* price - цена
* title - закодированное название товара (в файле vectors.npz содержатся эмбединги оригинальных названий)
* old\_price - старая цена

train\_actions.pq

* user\_id - глобальный идентификатор пользователя для подбора рекомендаций (формируется по cookies. Один пользователь может заходить с разных сайтов или устройств и иметь разные ID)
* loc\_user\_id - локальный идентификатор пользователя внутри сайта
* action - код действия пользователя
* date - дата и время действия
* products - массив идентификаторов товаров, относящийся к действию
* pageId - идентификатор страницы веб-сайта, с которой пользователь выполняет действие

catalog\_vector\_mapping.pq

* product\_id - идентификатор товара
* vector\_id - идентификатор вектора содержащего эмбединг названия товара из файла vectors.npz

vectors.npz - файл содержащий эмбединги названий товаров.

**Признак action в train\_actions.pq имеет следующую расшифровку:**

* 0 - view (просмотр товара)
* 1 - like (лайк товара)
* 2 - addB (добавление товара в корзину)
* 3 - delB (удаление товара из корзины)
* 4 - clearB (удаление всех товаров из корзины)
* 5 - order (оформление заказа)
* 6 - listB (посещение страницы корзины и вывод списка товаров в корзине)
* 7 - visit (посещение страницы магазина)
* 8 - visitCategory (посещение страницы с группой товаров)
* 9 - search (поиск товара)

**Общие требования:**

- Эксперимент, его результирующие артефакты и модель должны логироваться в MLFlow

- Код (пайплайн) предобработки данных и обучения модели должен быть разбит на отдельные скрипты, запускающиеся с помощью одной команды в нужной последовательности. Можно реализовать через собственный Python или консольный скрипт. Предпочтительно использовать workflow manager использующий DAG (snakemake или DVC)

- Код должен быть написан в соответствии с PEP8

- Зависимости проекта должны находиться под управлением одного из известных менеджеров и должны обеспечивать воспроизводимость эксперимента (pip, poetry)

- Репозиторий должен быть обеспечен подробной инструкцией по запуску эксперимента в Readme файле

**Состав решения:**

* Презентация решения должна отражать
  + Описание исходных данных
  + Используемый технологический стек и инструменты
  + Методологию проведения экспериментов
  + Методы обработки исходных данных
  + Основные гипотезы, проверяемые в ходе экспериментов
  + Описание проверяемых алгоритмов/архитектур
  + Подбор гиперпараметров
  + Результаты валидации моделей
* Выгрузка предсказаний модели на 3 дня в формате, представленном в файле baseline.ipynb
* Репозиторий с Readme (на Github/Gitlab), включающий:
  + Инструкции по развертыванию и установке решения для инференса (если требуется)
  + Инструкции по запуску пайплайнов обработки даннных и обучения модели
* Модель и артефакты эксперимента в MLFlow или другом registry для хранения моделей