Дипломная работа «Data Science: рекомендательные системы»

Ишмухаметов Айдар
DIP-DSR-CP-1

Постановка задачи

1

Описание

Исходная задача:

Создание рекомендательной системы для предоставления пользователям онлайн-кинотеатра персонализированных рекомендаций, основанных на их предпочтениях.

Актуальность задачи, ее место в предметной области:

Рост популярности и количества онлайн-сервисов предоставления видеоконтента приводит к необходимости создания качественных рекомендаций для удержания пользователя.

Целевая метрика:

map@K - Mean average precision at K. Отношение количества релевантных объектов к общему количеству рекомендаций с учетом ранжирования, усредненное по всем пользователям.

Анализ и подготовка данных

2

Анализ и подготовка данных

Данные получены с площадки RecSys Course Competition — Open Data Science. Датасет содержит данные из приложения МТС Kion по взаимодействиям пользователей с контентом за период 6 месяцев:

- факты просмотра контента пользователями
- описание контента
- описание пользователей

Размер датасета:

- Кол-во пользователей всего - 1 058 088
- Кол-во объектов всего - 15 963

Анализ и подготовка данных

Был проведен разведочный анализ данных. Проведена работа с пропусками и дубликатами.

Типы данных некоторых признаков были изменены с целью оптимизации размера занимаемой памяти.

Все имеющиеся данные были поделены на две части. Для этого все записи датасета были отсортированы по дате просмотра, в обучающую выборку попали 70%, а в тестовую 30% данных.

Методика реализации

3

Каким образом решали задачу?

Реализовано два типа рекомендаций для пользователя:

- фильмы, похожие на просмотренный
- рекомендация фильмов в соответствии с предпочтениями пользователя.

В зависимости от наличия данных реализовано три способа получения рекомендаций:

- Есть информация по просмотрам пользователя. Используются алгоритмы машинного обучения для получения рекомендаций.
- Есть только соц. характеристики пользователя (пол, возраст, наличие детей) и нет просмотров. Используется статистика - наиболее популярные фильмы по соц. группам
- По пользователю нет никакой информации. Используется топ самых популярных фильмов в качестве рекомендации.

Каким образом решали задачу?

Для создания рекомендаций была выбрана коллаборативная фильтрация. Ее преимущества:

- не нужно подготавливать фичи
- есть достаточно много данных, причем пользователей гораздо больше, чем фильмов
- данный способ является менее затратным с точки зрения перерасчета модели при добавлении новых объектов, так новые фильмы добавляются реже, чем пользователи

Для решения задачи был использован модуль AlternatingLeastSquares из библиотеки implicit (https://benfred.github.io/implicit/api/models/cpu/als.html). Полученная модель может рекомендовать фильмы для пользователя, а также предлагать похожие фильмы.

Каким образом решали задачу?

В центре рекомендательной системы находится так называемая матрица предпочтений (по одной из осей которой находятся все пользователи, а по другой – фильмы. На пересечении некоторых пар - известные показатели заинтересованности пользователя.

В качестве такой оценки использовался неявный признак - % просмотра фильма по длительности. Можно предположить, что если пользователь посмотрел фильм полностью (значение близкое к 100%), то он ему понравился. Если значение ближе к нулю, то скорее всего, фильм ему не понравился. Остается, конечно, вероятность что его отвлекли от просмотра и он уже к нему не вернулся.

Каждый пользователь, естественно, просмотрел лишь небольшую часть фильмов. И задача рекомендательной системы предсказать отношение клиента к другим фильмам, про которые пока ничего не известно. Другими словами нужно заполнить все незаполненные ячейки в матрице item-user.

Каким образом решали задачу?

Примеры похожих фильмов, полученных при помощи модели:

```
['железный человек']
Похожие фильмы:
железный человек..... score  1.0000001
железный человек 2..... score  0.9979548
железный человек 3..... score  0.9976278
первый мститель..... score  0.9883371
тор..... score  0.9842844
первый мститель: другая война..... score  0.974221
человек-муравей..... score  0.9709954
тор: царство тьмы..... score  0.9695387
первый мститель: противостояние..... score  0.9487252
лондон убивает..... score  0.94860816
```

Каким образом решали задачу?

Также была реализована content-based рекомендательная система. Был проведен feature engineering (tf-idf с лемматизацией на таких признаках как жанры, страны, режиссеры, актеры и описание фильма).

Плюсом этой системы является простота: рекомендуются фильмы похожие на те, которые пользователь уже посмотрел. Но, в то же время, это является и минусом: пользователю не предлагается новое.

Примеры похожих фильмов:

```
['железный человек']
Похожие фильмы:
['железный человек']..... score  0.0
['железный человек 2']..... score  0.7276383214825564
['человек-паук: возвращение домой']..... score  0.8209425896736033
['железный человек 3']..... score  0.8256363076102233
['план побега']..... score  0.883161101635186
['мстители: финал']..... score  0.8942020359529614
['повар на колесах (жестовым языком)']..... score  0.9139904874428508
['восстание']..... score  0.9174916598834928
['мордекай']..... score  0.9179710187865464
['мстители']..... score  0.9182717867927455
```

Итоги и выводы

4

Описание итоговой модели

Полученная рекомендательная система позволяет предлагать пользователю фильмы, похожие на те, что он посмотрел (например, когда он открывает страницу с фильмом или сразу после просмотра фильма показывается блок “фильмы похожие на этот”) и рекомендовать фильмы, не связанные с конкретным (например, блок “что еще можно посмотреть”)

Предусмотрен “холодный старт” для пользователей - предлагается топ-10 фильмов в зависимости от имеющихся данных по пользователю (пол, возраст, наличие детей)

Рекомендательная система показала метрику качества MAP@10 = 0.105 на тестовых данных, что сравнимо с показателями метрики в лидерборде на странице соревнования https://ods.ai/competitions/competition-recsys-21/leaderboard

Дальнейшие пути развития и улучшения системы. Изучить другие алгоритмы рекомендательных систем, попробовать скомбинировать рекомендации, полученные несколькими способами, таким образом повысить качество системы в целом.

Спасибо за внимание!