Применение алгоритмов коллаборативной фильтрации для рекомендации банковских услуг

Романцова Вера
Студент группы DS-17

Постановка задачи

1

Рекомендательные системы (RS)

- это комплекс алгоритмов, программ и сервисов, задача которого предсказать, что может заинтересовать того или иного пользователя на основе знаний о пользователе, продукте и его предыдущих покупок или интересов.

https://vc.ru/marketing/152926-rekomendatelnye-sistemy-kak-pomoch-polzovatelyu-nayti-to-chto-emu-nuzhno

Описание проблемы

В условиях переизбытка информации пользователем крайне важно рекомендовать ему то, что будет наиболее интересным в текущий момент времени.

Рекомендательная система для банковской сферы отличается от классических рекомендательных систем, таких как Netflix, Amazon и пр. тем, что:

1. Информация о банковских продуктах зачастую очень скудна.

2. В традиционных задачах RS продукту присваивается рейтинг, как правило, по 5-10 балльной шкале. У банков же есть только бинарная информация о покупке продукта.

3. Самих банковских продуктов немного.

4. Но при этом, даже когда человек еще не стал клиентом банка, а только думает об этом, банк уже обычно знает о человеке очень много.

Актуальность проблемы

Наличие рекомендательной системы позволяет компании:

1. Увеличить прибыль
2. Увеличить продажи
3. Увеличить жизненную ценность клиента
4. Увеличить лояльность пользователей

По оценкам McKinsey, благодаря рекомендательным системам, Amazon увеличивает продажи на 35%, а Netflix - на 75%.

mckinsey.com/industries/retail/ our-insights/how-retailers-cankeep-up-with-consumers

Этапы решения задачи

1. Первичный анализ задачи.
2. Поиск аномалий и заполнение пропусков в данных.
3. Создание base-line модели - item-based коллаборативной фильтрации, подбор гиперпараметров.
4. Создание альтернативной модели на основе скрытых факторов (на основе SVD)
5. Решение проблемы холодного старта для нового пользователя.
6. Создание более сложной гибридной модели.
7. Анализ качества моделей.

Целевая метрика

В основе рекомендательной системы лежит получение прогноза оценки продукта потребителем.

Мы будем сравнивать прогнозное значение оценки с фактическим. В нашем случае оценка бинарная: купил/не купил (0/1).

Метрика для сравнения модели и подбора гиперпараметров:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Первичный анализ

2

EDA
Исходные данные

Был использован датасет, содержащий покупки банковских продуктов крупнейшей финансово-кредитной группы Испании Santander
Ноутбук доступен по ссылке.

- Объем датасета:
  13 647 309 объектов. Каждый объект - история покупок пользователем за месяц. Всего в выборке N месяцев на каждого пользователя
- Количество уникальных пользователей:
  956 645. Для пользователей в датасете указаны характеристики пользователя, н-р такие как пол, возраст, уровень дохода и пр. Всего характеристик - 21.
- Количество продуктов:
  24

*Ввиду того, что датасет большой и для расчетов требуются сервера с высокой производительностью для расчетов использовался датасет по 2% пользователей: 19132

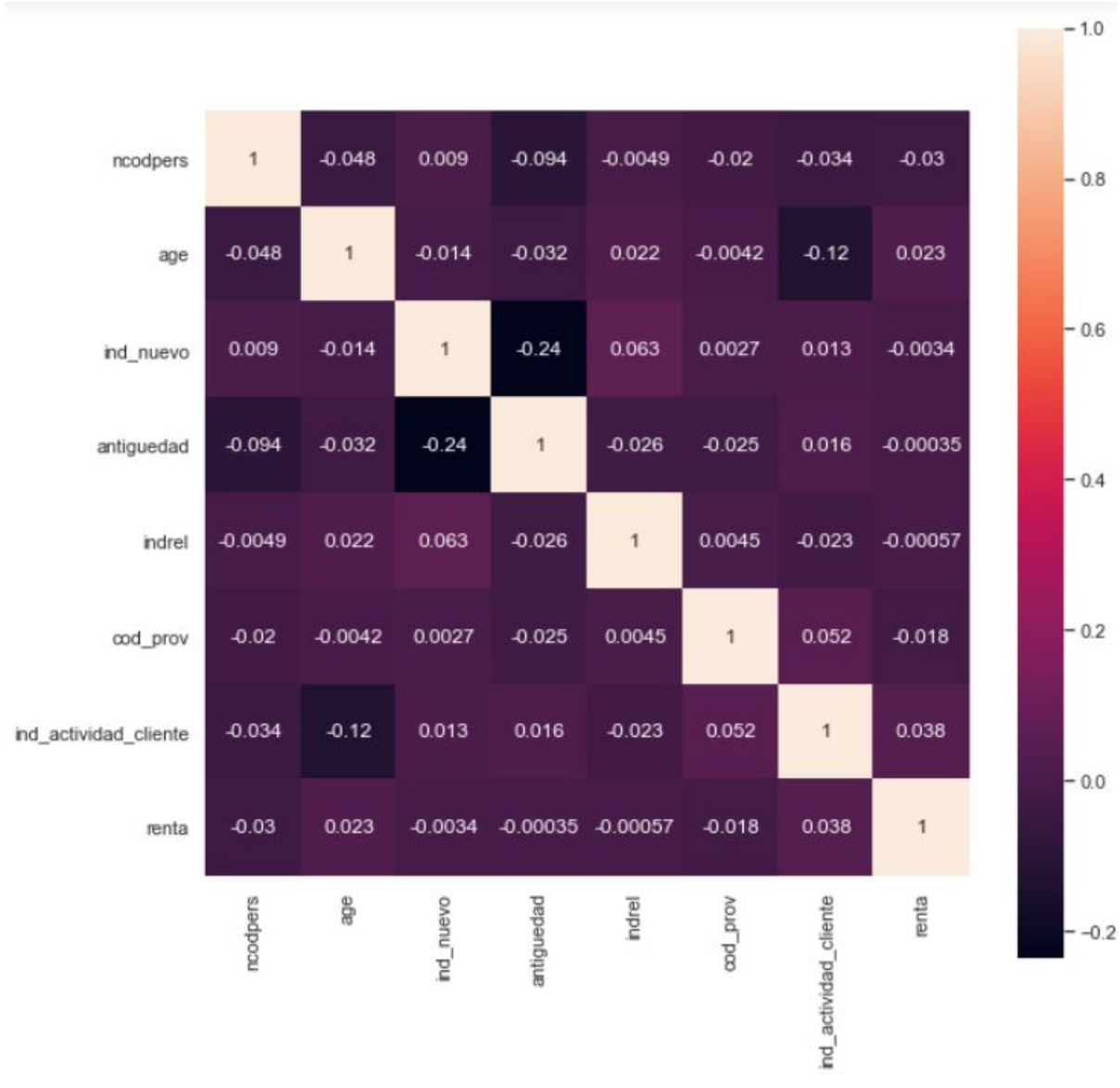
https://www.kaggle.com/c/santander-product-recommendation

EDA
Выбор признаков

Удалим признаки из датасета, которые фактически дублируют информацию:

- например, код провинции и название провинции

Количественные признаки
не скоррелированы друг с другом
 - можем их использовать все



Поиск аномалий и работа с пропусками

3

EDA
Работа с пропусками

- <u>Для признаков по потребителю:</u>

Для категориальных признаков заменили пропуски значением моды по каждому признаку.

Пропуски по числовым признакам заменили средним значением по каждой провинции*

- <u>Пропуски по продуктам:</u>

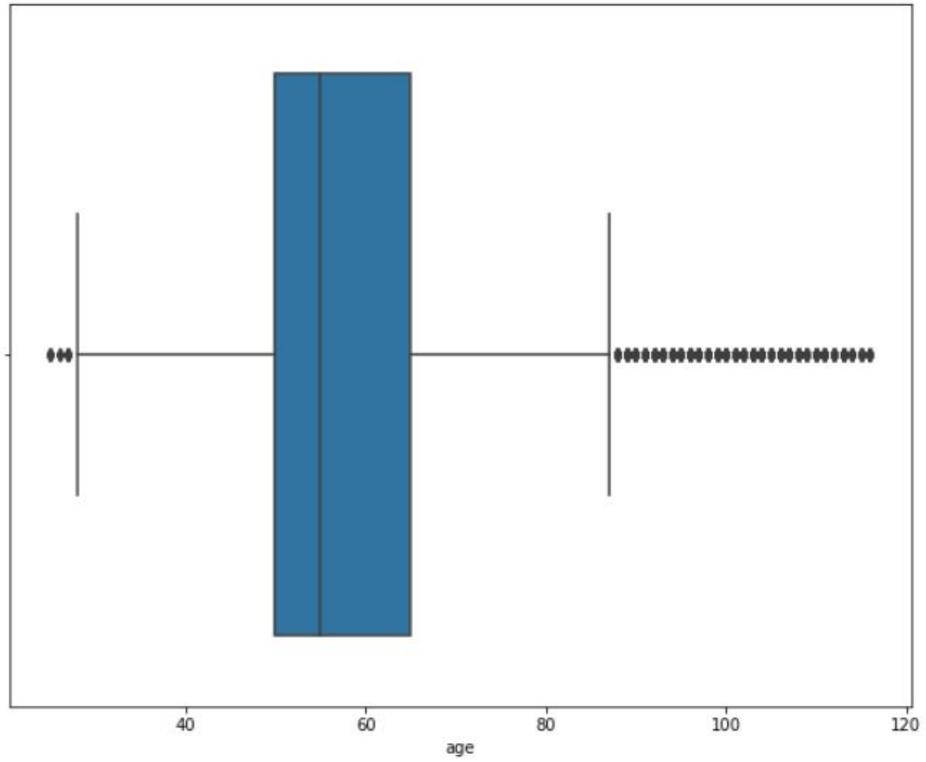
Заменили на 0, так если данных о приобретении продукта нет, мы не можем говорить о его приобретении.

fecha_dato	0
ncodpers	0
ind_empleado	764
pais_residencia	764
sexo	764
age	764
fecha_alta	764
ind_nuevo	764
antiguedad	764
indrel	764
indrel_1mes	765
tiprel_1mes	765
indresi	764
indext	764
conyuemp	290501
canal_entrada	3695
indfall	764
tipodom	765
cod_prov	2156
ind_actividad_cliente	764
renta	39575
segmento	4028

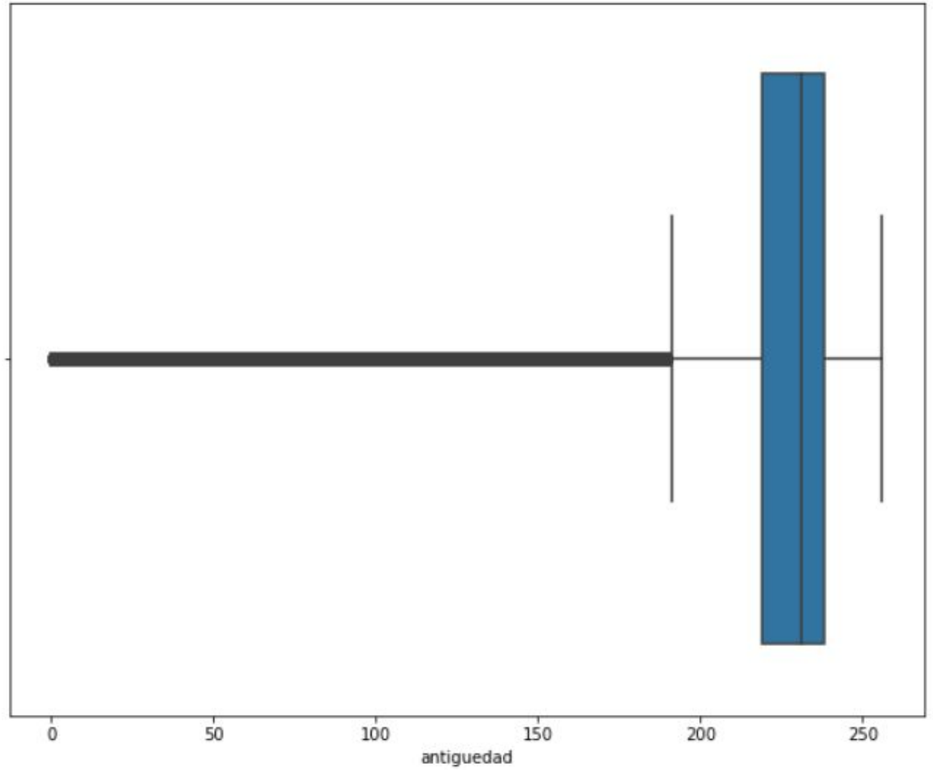
- в рамках урезанного датасета часть признаков пришлось заменить значениями по всему датасету

EDA
Поиск аномалий

- Для количественных переменных данные проверены на наличие аномалий



Долгожителей не убираем из датасета - для них также есть продукты - пенсии



Стаж работы покупателей также в пределах разумного и не превышает 20 лет

Методы и алгоритмы

4

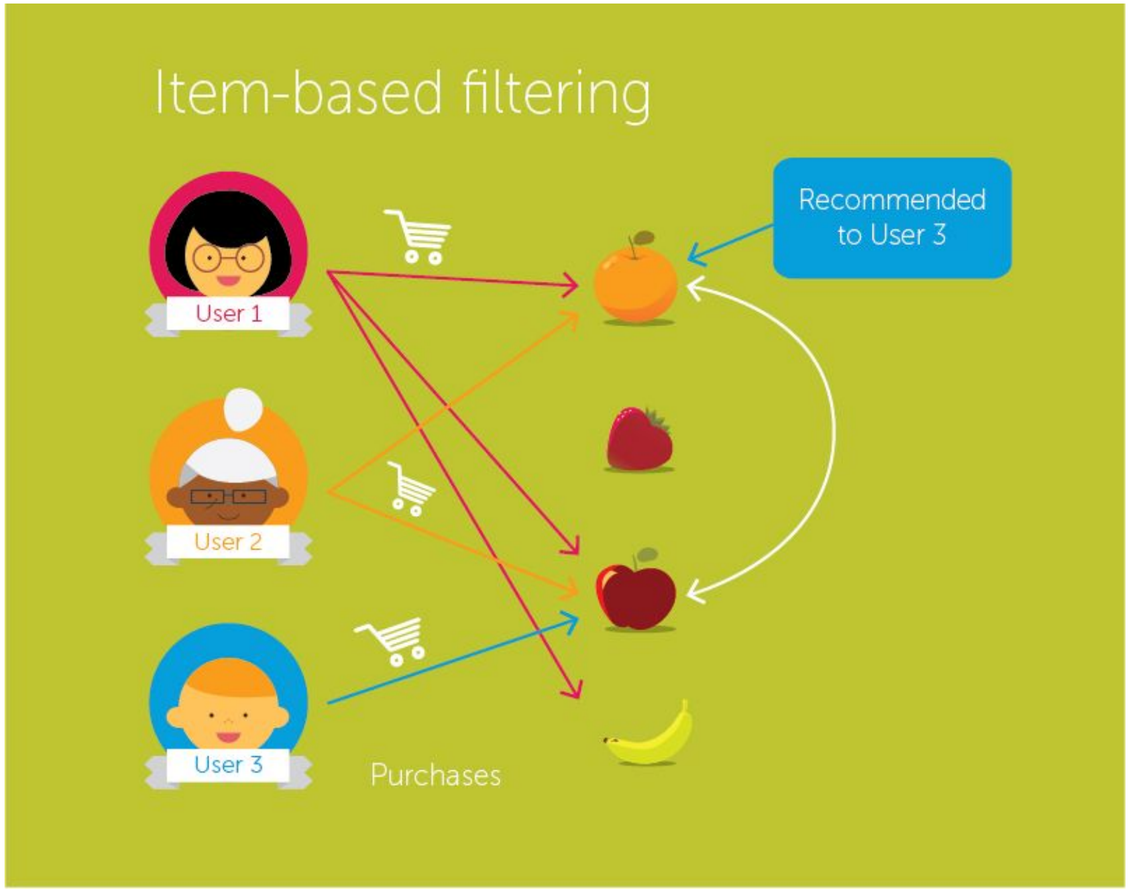
Item-based Collaborative Filtering

\+ Преимущества

- универсальность метода
- вычислимость (если продуктов меньше, чем пользователей)

\- Недостатки

- проблема “холодного” старта

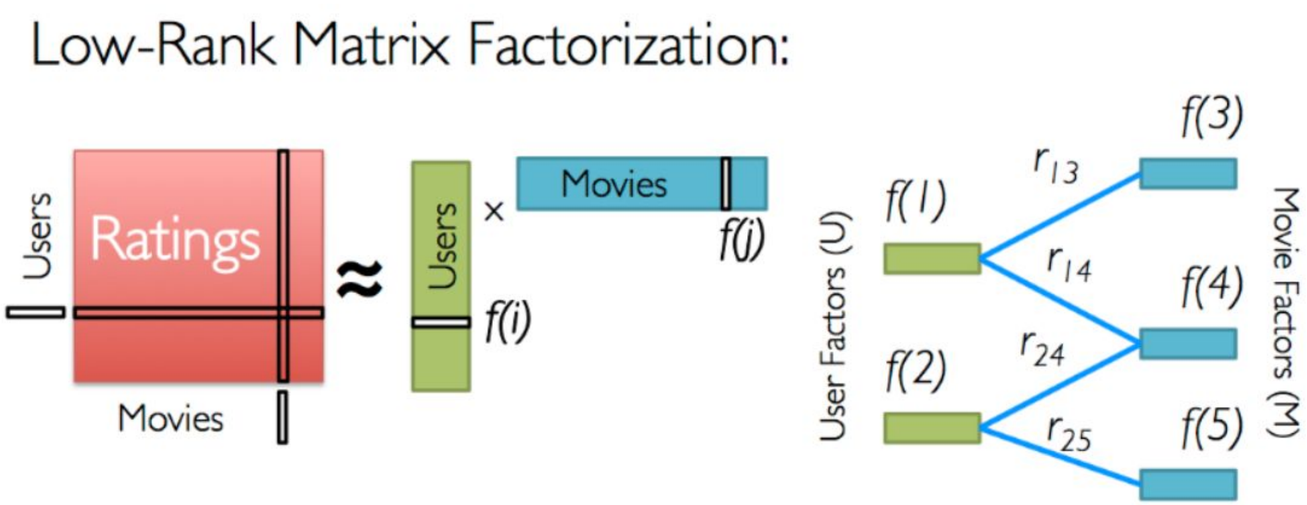


https://habr.com/ru/company/lanit/blog/421401/

SVD-разложение (рекомендации на основе скрытых факторов)

+ Преимущества

- алгоритм позволяет выявлять скрытые признаки объектов и интересы пользователей



- Недостатки

- матрица Ratings нам полностью неизвестна и не можем просто взять ее SVD разложение
- SVD разложение нужно пересчитывать с каждым новым пользователем и товаром
- вычислимость

https://habr.com/ru/company/lanit/blog/421401/

http://neerc.ifmo.ru/wiki/index.php?title=%D0%A0%D0%B5%D0%BA%D0%BE%D0%BC%D0%B5%D0%BD%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B

Hybrid RS

\+ Преимущества

- объединяют несколько подходов и имеют наибольшую эффективность

\- Недостатки

- наиболее сложны в реализации
- вычислимость

*Например, у Netflix в гибридной системе объединено 27(!) алгоритмов

https://vc.ru/marketing/152926-rekomendatelnye-sistemy-kak-pomoch-polzovatelyu-nayti-to-chto-emu-nuzhno

Аналоги

1. XGBoost
2. RandomForestClassifier

https://vc.ru/marketing/152926-rekomendatelnye-sistemy-kak-pomoch-polzovatelyu-nayti-to-chto-emu-nuzhno

Этапы реализации

5

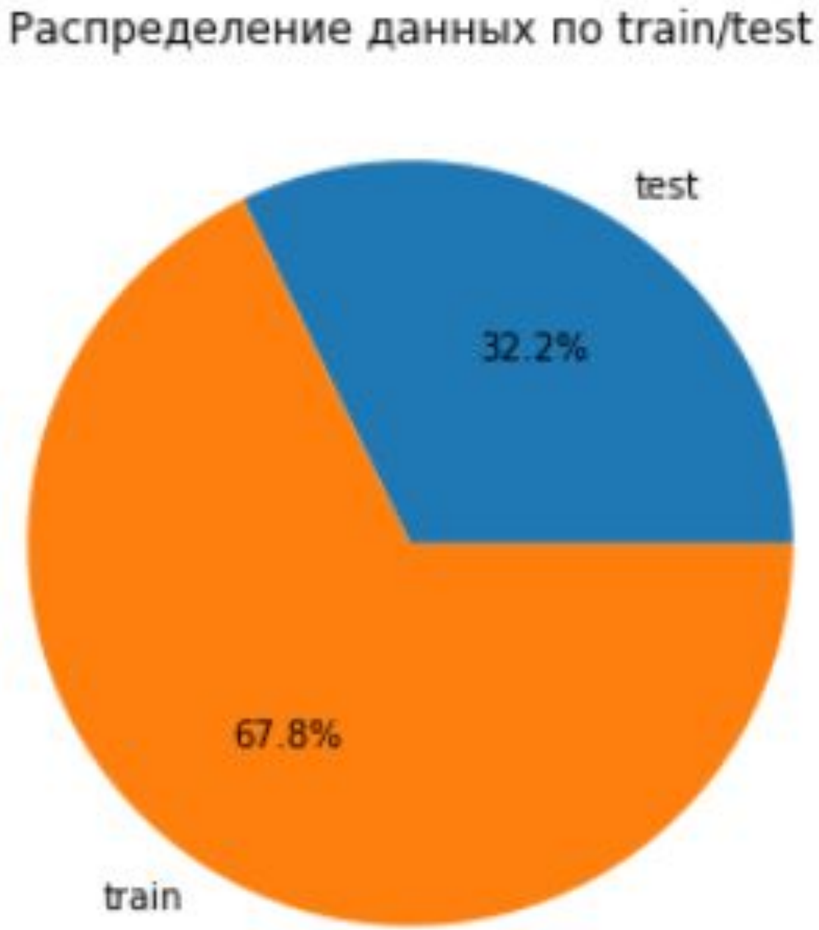
Этапы реализации системы рекомендации

1. Будем обучаться на данных до 2015-12-28, остальное идет в тест.
2. Строим base-line: item-based CF
3. Создание альтернативной модели на основе SVD
4. Решение проблемы “холодного” старта
5. Создание гибридной модели, сравнительный анализ метрик

Разбивка датасета на train/test

- в исходном датасете по пользователю известны данные о нем (пол, возраст и т.д)
- в классических RS используется только 3 признака (user_id, product_id, rating)
- для корректного сравнения метрик классических методов RS и с использованием признаков пользователя делали разбивку по дате покупки

- был написан метод для разбивки данных на train/test с учетом этого фактора



Item-based Collaborative Filtering

На основе KNNWithMeans подобраны следующие гиперпараметры, минимизирующие RMSE:

- с количеством соседей, равным 10
- и мерой сходства: корреляцией по Пирсону

SVD-разложение (рекомендации на основе скрытых факторов)

На основе SVD использованы следующие гиперпараметры:

- с числом скрытых факторов, равным 20
- и на 20 эпохах

Решение проблемы “холодного” старта Cold-start

На основе характеристик пользователя (пол, возраст и пр.) на основе сходства признаков найти продукты, наиболее интересные “соседям”:

- нашли пользователей с максимальным количеством категориальных совпавших фич
- получили рейтинги “соседей” для разных продуктов
- получили финальные рекомендации усреднением рейтингов “соседей”

Создание гибридной модели

Если пользователь ничего не купил:

- Рекомендуем ему то, что выбрали пользователи - “соседи” (с максимальным набором одинаковых признаков)

Если пользователь уже совершал покупки:

- Рекомендации на основе Item-based CF, SVD, Cold-start в равной пропорции

Результаты

5

Сравнительный анализ метрик и выбор модели

- Для случайных 1000 пользователей из тестовой выборки предсказали рекомендации и сравнили с ground_truth

Model	RMSE ± std
RMSE for Cold_Start	0.1235 ± 0.0510
RMSE for Item-based CF	0.1427 ± 0.0614
RMSE for SVD	0.1387 ± 0.0637
RMSE for Hybrid	**0.1161 ± 0.0506**

Гипотеза, почему так происходит: (Cold-start лучше Item-based CF) - т.к. в обоих методах используется поиск ближайших соседей, то одна из возможных причин такого поведения - это намного большая информативность категориальных фичей при поиске ближайшего соседа, именно поэтому Cold-start используется в гибридной модели для пользователя с покупками.

Выводы

6

Выводы:

1. В ходе работы проанализирован исходный датасет, данные проверены на наличие аномалий и заполнены пропуски.

2. Для построения рекомендательной системы выбраны Item-based CF, рекомендации на основе скрытых факторов, гибридная модель и решение проблемы холодного старта на основе близости признаков пользователей.

3. В ходе работы выяснено, что категориальные признаки пользователя можно использовать для решения проблемы холодного старта, и помимо этого, использовать для пользователей, уже совершавших покупки ранее.

4. По данным RMSE выбрана лучшая модель - **Hybrid model**

5. Написана функция, дающая рекомендации для пользователя из датасета.

Дальнейшие пути развития:

1. На сервере подобрать оптимальные параметры для SVD
2. Использовать более сложную модель для холодного старта (аля XGBoost)
3. Подобрать наилучшие веса для ансамбля моделя в Hybrid
4. На высокопроизводительном сервере использовать данные всего датасета

Спасибо за внимание!