Collaborative Filtering

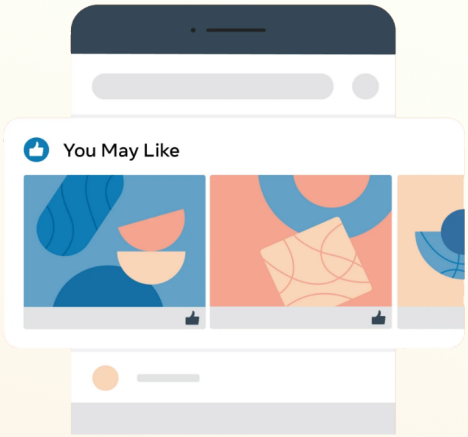
ARIEL UNIVERSITY

Presenter: Shaked Shiran

Instructors: PhD Anat Goldstein, PhD Chen Hajaj

1 Project goal

Develop a model that provides recommendations to customers on E-commerce websites based on other customers' browsing data



2 Introduction

E-commerce websites sell thousands of different products and require tools to customize the products displayed for the customer
Most e-commerce websites display a list of recommended products for the customer.
The suitability of the offered products in the display to the customer is important for the website, which has limited advertising space and wants the customer to spend time on the websites and being exposed to products he may purchase.
And a customer, who prefers to browse websites that provide him a sense of suitability for him and his needs.

3 Selected Approach

The selected approach in the project is based on developing collaborative filtering models and then comparing them to other similarity models to produce a hybrid model that provides an accurate list of recommendations to the customer according to his recent searches on the website.

4 Collaborative filtering

The Collaborative Filtering model provide to predict the user's chances of liking destinations, based on the browsing data of other users who have been interested in similar destinations.

5 Database description

The database contains 600,000 customer sessions on a flight search site.
Each session indicates the customer's ID, the destinations he was looking for, and whether he placed an order or not

Session Id	CITY
Session 1	1, 2, 1, 2, 2, 4
Session 2	2
Session 3	3, 1, 3
Session 4	5
Session 5	5, 3, 5, 3, 5

→

Session Id \ City	CITY 1	CITY 2	CITY 3	CITY 4	CITY 5
Session 1	2	3	0	1	0
Session 2	0	1	0	0	0
Session 3	1	0	2	0	0
Session 4	0	0	0	0	1
Session 5	0	0	2	0	3

6 Solution Description

The algorithm was developed in Jupiter's notebook in Python language.
The raw database is converted to Data-frame of the Pandas library.
transformation is applied on the Data-frame to create a utility matrix that contains the city IDs as columns, and for each city its number of searches in each session is specified.

Different strategies are applied on the Utility-Matrix to find list of recommended destinations for the customer.

We will present two different methods:

Method 1

In this method we check the number of common destinations for the customer's searches and for each of the sessions.
After filtering the sessions that contain similar searches, we summarize for each city the number of times it was searched.
This allows us to know which cities the users searched for the most along with the customer's searches and get a sorted list of recommendations for the customer.

Session Id \ City	CITY 1	CITY 2	CITY 3	CITY 4	CITY 5	Similarity
Session 1	2	3	0	1	0	1 + 0 + 1 = 2
Session 2	0	1	0	0	0	0 + 0 + 0 = 0
Session 3	1	0	2	0	0	1 + 1 + 0 = 2
Session 4	0	0	0	0	1	0 + 0 + 0 = 0
Session 5	0	0	2	0	3	0 + 1 + 0 = 1

User searches:

{ 4, 4, 3, 1 }

CITY 1	CITY 2	CITY 3	CITY 4	CITY 5
1	0	1	2	0

Method 2

In this method we refer to the number of searches for each destination.
In this similarity function, for each destination the customer searched for, the number of times the customer searched for; it was multiplied by the number of times the destination was searched in each session. The amount received from this multiplication for each of the customer's searches provides an estimate of the session's similarity to the customer's searches.
Sessions sorted by their level of similarity make it possible to get a sorted list of recommendations.

Session Id \ City	CITY 1	CITY 2	CITY 3	CITY 4	CITY 5	Similarity
Session 1	2	3	0	1	0	2*1 + 3*0 + 0*1 + 1*2 + 0*0 = 4
Session 2	0	1	0	0	0	0*1 + 1*0 + 0*1 + 0*2 + 0*0 = 0
Session 3	1	0	2	0	0	1*1 + 0*0 + 2*1 + 1*2 + 0*0 = 5
Session 4	0	0	0	0	1	0*1 + 0*0 + 0*1 + 0*2 + 1*0 = 0
Session 5	0	0	2	0	3	0*1 + 0*0 + 2*1 + 0*2 + 3*0 = 2

7 Statistics

To test the effectiveness of the model, we omitted the last destination in each session and tested the algorithm's success by running the algorithm for each session and checking whether the last destination appears in the list of recommendations for the session

Visit us on GitHub