Information Retrieval Baseline Results

“Product Recommendation System”

Group No. 28

- Tarun Kumar Gupta
- Shubham Sethi
- Kartik Gupta
- Rachit Gupta
- Akshat Tilak
- Ritika Nagar

With the wide variety of products and services available on the web, it is difficult for users to choose the product or service that most meets their needs. To reduce or even eliminate this difficulty, recommender systems have emerged. Recommendation systems are a particular type of information filtering system. They are software applications that aim to support users in their decision-making while interacting with a vast amount of information.

GAPS THAT NEED TO BE FILLED IN CURRENT RECOMMENDATIONS SYSTEMS

The current product recommendation system suffers from two significant drawbacks:

- Recommendation redundancy.
- Unpredictability concerning new items (cold start).

These limitations occur because the legacy recommendation systems rely only on the user’s previous buying behavior to recommend new items.

UPDATED PROBLEM STATEMENT

The proposed recommendation systems have already solved the problem of cold start by giving generalized recommendations to that user based on the popularity of a particular item by analyzing whatever is more popular among the general public that is more likely to be recommended to new customers.
Now the next task in our project is we will implement the product recommendation based on the like and dislike of the different users to a particular user because he/she will also most likely to purchase this product only.

LITERATURE REVIEW

According to the study [1], in this era of the web, we have a huge amount of information overload over the Internet. It becomes a big task for the user to get relevant information. To some extent, search engines are solving the problem, but they need to provide the personalization of data. Recommender system algorithms are widely used in e-commerce to provide personalized and more accurate recommendations to online users and enhance the sales and user stickiness of e-commerce. This study aims to build a product recommendation system on an e-commerce platform according to user needs.

The report "Collaborative Filtering for Recommender Systems Publication: 2014 Second International Conference on Advanced Cloud and Big Data" by Michael D. Ekstrand, John T. Riedl, and Joseph A. Konstan: highlights the discussion of the types of recommender systems as general and types of CF such as; memory-based, model-based, and hybrid models. In addition, this report discusses how to choose an appropriate type of CF. The evaluation methods of the CF systems are also provided throughout the paper. Features there are several limitations for the memory-based CF techniques, such as the similarity values being based on common items and unreliable when data are sparse, and the common items are few. Model-based CF approaches have been investigated to achieve better prediction performance and overcome the shortcomings of memory-based CF algorithms.

Generally speaking, product recommendation systems are divided into two main classes: one of them is [2] Collaborative filtering (CF); CF systems recommend new products to a given user based on their previous (rating/viewing/buying) history. Content-based recommender systems make recommendations by analyzing the content of textual information and finding regularities in the content. Also referred to as cognitive filtering, recommends items based on a comparison between the content of the items and a user profile. The content of each item is represented as a set of descriptors or terms, typically the words that occur in a document. According to Vishal Tomar and Aditya Kathuria [3] CBF uses different types of models to find similarities between documents to generate meaningful recommendations. Content-based filtering techniques do not need the profile of other users since they do not influence recommendations. A study [2] suggested that the major difference between CF and content-based recommender systems is that CF only uses user-item rating data to make predictions and recommendations. In contrast, content-based recommender systems rely on the features of users and items for predictions. Both content-based recommender systems and CF systems have limitations. While CF systems do not explicitly incorporate feature information, content-based systems do not necessarily include the information in preference similarity across individuals. Collaborative filtering models are based on the assumption that people like things similar to other things they want and things that other people with similar tastes like.

BASELINE RESULTS

Data Preprocessing

The first step in the project is to preprocess the suitable dataset for the product recommendation system, by dropping the column "Timestamp" and "Unnamed 0 ".
We use the library panda for preprocessing. Further, we preprocess the data according to the different submodels used in our project.

Sub-Models

Popularity-based recommendation system

In this product model, we recommend products to the user according to the highest mean average rating of products. In this, we only consider the rating of those users who have rated at least 5(a threshold that can change) different products. It does not suffer from cold start problems.

Content-based recommendation system

In this product model , we recommend products to the user based on Cosine similarity or a euclidean distance of the vector of two products(item).
We find cosine similarity and euclidean distance with the help of pivot table using sklearn library.

Output

Cosine similarity table

```python
similarity_score = cosine_similarity(pt)
df2 = pd.DataFrame(similarity_score)
df2
```

	0	1	2	3	4	5	6	7	8	9	...	3081	3082	3083	3084	3085	3086	3087	3088	3089	3090
0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.707107	...	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	1.0	0.707107	...	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.707107	...	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
3086	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.000000	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
3087	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.131306	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
3088	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
3089	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
3090	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

3091 rows × 3091 columns

Euclidean distance table

```python
similarity_score_2 = euclidean_distances(pt)
df3 = pd.DataFrame(similarity_score_2)
df3
```

Python

	0	1	2	3	4	5	6	7	8	9	...	3081	3082	3083	30
0	0.000000	5.099020	5.099020	5.099020	5.099020	5.099020	5.099020	7.141428	5.099020	7.141428	...	7.681146	3.162278	12.845233	4.1231
1	5.099020	0.000000	7.071068	7.071068	7.071068	7.071068	7.071068	8.660254	7.071068	8.660254	...	9.110434	5.830952	13.747727	6.4031
2	5.099020	7.071068	0.000000	7.071068	0.000000	7.071068	7.071068	8.660254	7.071068	5.000000	...	9.110434	5.830952	13.747727	6.4031
3	5.099020	7.071068	7.071068	0.000000	7.071068	7.071068	0.000000	8.660254	0.000000	5.000000	...	9.110434	5.830952	13.747727	6.4031
4	5.099020	7.071068	0.000000	7.071068	0.000000	7.071068	7.071068	8.660254	7.071068	5.000000	...	9.110434	5.830952	13.747727	6.4031
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
3086	5.000000	7.000000	7.000000	7.000000	7.000000	7.000000	7.000000	8.602325	7.000000	8.602325	...	9.055385	5.744563	13.711309	6.3245
3087	1.414214	5.099020	5.099020	5.099020	5.099020	5.099020	5.099020	7.141428	5.099020	7.141428	...	7.549834	3.162278	12.845233	4.1231
3088	11.313708	12.328828	12.328828	12.328828	12.328828	12.328828	12.328828	13.304135	12.328828	13.304135	...	13.601471	11.661904	17.058722	11.9582
3089	5.099020	7.071068	7.071068	7.071068	7.071068	7.071068	7.071068	8.660254	7.071068	8.660254	...	9.110434	5.830952	13.747727	6.4031
3090	8.185353	9.539392	9.539392	9.539392	9.539392	9.539392	9.539392	10.770330	9.539392	10.770330	...	11.135529	8.660254	15.165751	9.0553

3091 rows × 3091 columns

REFERENCES

[1] Xuesong Zhao, " Study of an E-commerce recommender system based on Big data Publication.", Publication: Oxbridge college, kunming university, Available:https://ieeexplore.ieee.org/document/8725694

[2] Khartoum and Sudan, "Content-Based Filtering: Techniques and Applications.", Publication: 2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE),
Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7867668

[3] Parul Aggarwal, Vishal Tomar, Aditya Kathuria, "Comparing content-based and collaborative filtering in recommendation systems",
Available:
https://www.neliti.com/publications/263309/comparing-content-based-and-collaborative-filtering-in-recommender-systems