Movie Recommender System

Group 1 Presentation
By
Swetha Kalla, Tanvi Hindwan, Jyoti Sharma

Outline

- Introduction to Recommender System
- Types of Recommender System
- Dataset Details and Data Preprocessing
- Content Based - Recommender System
- Drawbacks of Content Based - Recommender System
- Item Based - Collaborative Filtering
- Advantages and Challenges of Item Based - Collaborative Filtering
- Future Work
- References

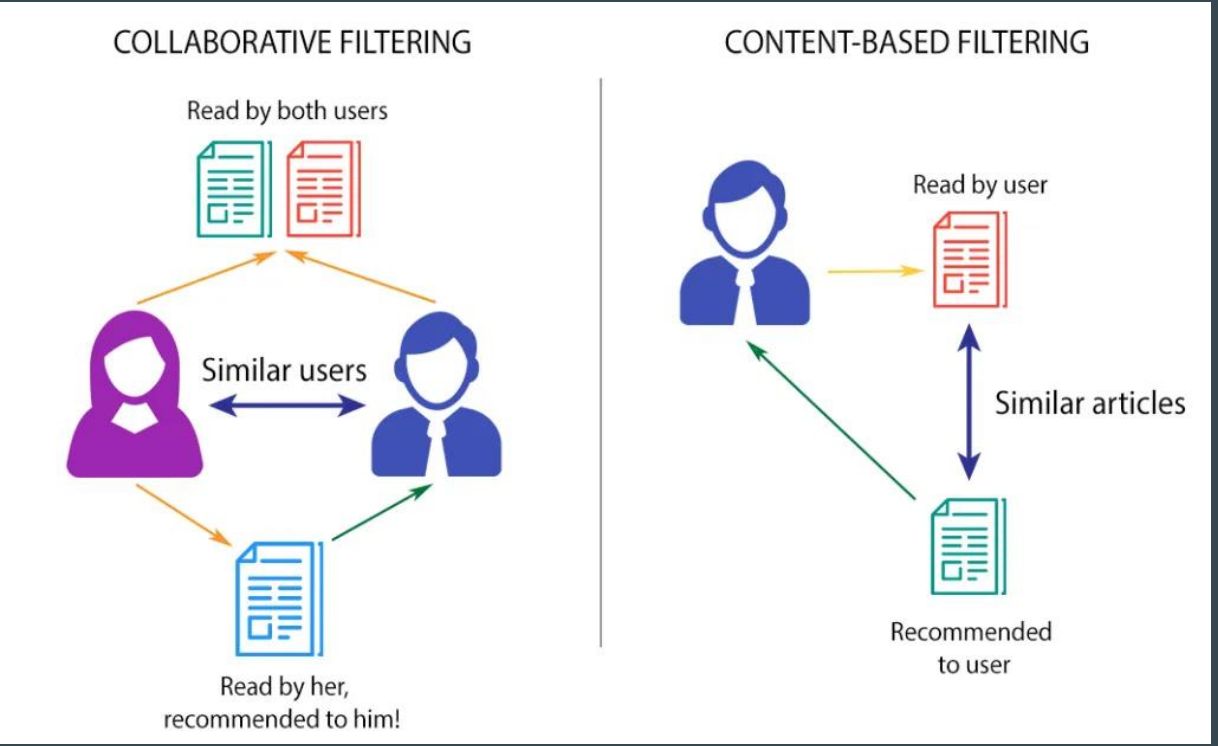
What is a Recommender System?

Recommender systems are algorithms aimed at suggesting relevant items to users. For example: movies to watch, text to read, products to buy, etc.

Youtube, Netflix, Amazon, Pinterest, and long list of other internet products all rely on recommender systems to filter millions of contents and make personalized recommendations to their users.

Recommender systems are crucial in some industries as they can generate a huge amount of income when they are efficient or also be a way to stand out significantly from competitors.

Types of Recommender System



http://datameetsmedia.com/an-overview-of-recommendation-systems/

Dataset Details

MovieLens *(Collaborative)* datasets were collected by the GroupLens Research Project at the University of Minnesota.

This data set consists of 100,000 ratings (1-5) from 943 users on 1682 movies. Each user has rated at least 20 movies.

The Movies Dataset *(Content Based)* This dataset is a collection of data from TMDB and GroupLens. The Movie Details, Credits and Keywords have been collected from the TMDB Open API.

Data Preprocessing

df ['vote_count'] > df['vote_count'].quantile(0.50)

df = df.dropna(axis = 0, how = ‘any’) *(Large dataset with extremely low NaN Values)*

extracted year of release from datetime format

The dataset has been reduced from 45,466 movies to 21,748 movies. *(Done also for the sake of computational power)*

Content Based - Recommender System

Content based systems provide recommendations based on the user profile and metadata it has on particular item.

The models compute the pairwise similarity of the text by using CountVectorizer() and the TF-IDF Vectorizer.

We used TF-IDF Vectorizer because: the CountVectorizer favours of most frequent words whereas TF-IDF Vectorizer considers the overall document weightage of a word by penalizing the most frequent words.

Based on the movie summary Metadata

```
content_recommender('Jumanji')
```

```
13912         Table No. 21
21707                 Quiz
6723               Quintet
4789             Brainscan
17401         Turkey Shoot
20086            Beta Test
11754                DeVour
4703      Poolhall Junkies
17115               Pixels
16658              Standby
Name: title, dtype: object
```

Based on

```
content_recommender('Jumanji', cosine_similarity_2, df, indices2)
```

```
10563                        Where the Wild Things Are
450                                       The Pagemaster
15348                 Tinker Bell and the Lost Treasure
17233    Mostly Ghostly: Have You Met My Ghoulfriend?
9832                                        City of Ember
17240                 Zenon: Girl of the 21st Century
9335                                      The Water Horse
15980                                          Snow Queen
19782                        The Shamer's Daughter
1511                                        Return to Oz
Name: title, dtype: object
```

Drawbacks

- Content based systems do not leverage the power of community: Results might not be as impressive as collaborative systems.
- Provides recommendations that are obvious.

Advantage

- Does not require a lot of data: Provides recommendations based on user profile and metadata.

Item based - Collaborative Recommender System

- Item-based Collaborative filtering works on the principle based on similarity between each pairs of items. Here, we will find similarity between each movie pair and based on that, we will recommend similar movies which are liked by the users in the past.

- Similarity Measure - Pearson Correlation based similarity

	user_id	movie_id	rating	title
0	196	242	3	Kolya (1996)
1	63	242	3	Kolya (1996)
2	226	242	5	Kolya (1996)
3	154	242	3	Kolya (1996)
4	306	242	5	Kolya (1996)

Dataframe overview for Item based

Item based - Collaborative Recommender System

user_id	'Til There Was You (1997)	1-900 (1994)	101 Dalmatians (1996)	12 Angry Men (1957)	187 (1997)	2 Days in the Valley (1996)	20,000 Leagues Under the Sea (1954)	2001: A Space Odyssey (1968)	3 Ninjas: High Noon At Mega Mountain (1998)	39 Steps, The (1935)
1	NaN	NaN	2.0	5.0	NaN	NaN	3.0	4.0	NaN	NaN
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1.0	NaN
3	NaN	NaN	NaN	NaN	2.0	NaN	NaN	NaN	NaN	NaN
4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
5	NaN	NaN	2.0	NaN	NaN	NaN	NaN	4.0	NaN	NaN

Sparse matrix with user ratings as input for given user and movie

title	'Til There Was You (1997)	1-900 (1994)	101 Dalmatians (1996)	12 Angry Men (1957)	187 (1997)	2 Days in the Valley (1996)	20,000 Leagues Under the Sea (1954)	2001: A Space Odyssey (1968)	3 Ninjas: High Noon At Mega Mountain (1998)	39 Steps, The (1935)
'Til There Was You (1997)	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1-900 (1994)	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
101 Dalmatians (1996)	NaN	NaN	1.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
12 Angry Men (1957)	NaN	NaN	NaN	1.0	NaN	NaN	NaN	0.178848	NaN	NaN
187 (1997)	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Correlation matrix with movie similarity score. **NAN** means no people are found in common who watched both movies

```
Similarities for movie:Lion King, The (1994)    5.0
Similarities for movie:Toy Story (1995)         5.0
Similarities for movie:Young Guns (1988)        1.0
```

Dummy User profile

```
Robin Hood: Prince of Thieves (1991)                  5.188938
Beauty and the Beast (1991)                           5.142471
Aladdin (1992)                                        4.838503
Winnie the Pooh and the Blustery Day (1968)           4.643764
Jungle Book, The (1994)                               4.607264
Fox and the Hound, The (1981)                         4.583560
Cool Runnings (1993)                                  4.419270
Cinderella (1950)                                     4.385974
Firm, The (1993)                                      4.380654
Mrs. Doubtfire (1993)                                 4.332134
dtype: float64
```

Similar Movies recommended to user

Key Advantage with Item based

- Item-based CF is quite stable, which means ratings on a given item does not change significantly over time.

Challenges with Item based

- It is required to have enough users to find the similar items (movies).
- Sparsity problem: Many times users don’t rate even they liked the item.
- Cold Start: Can not handle new items if the item like that is not rated previously.

Future Work

- The recommender system can be better implemented using PySpark as it leverages the ‘Big Data’.
- Building a Hybrid recommendation engine which is an ensemble of Content Based and Collaborative Filtering.
- Making use of Surprise - A simple python library for building and testing recommendation engines.

References

https://www.kaggle.com/rounakbanik/the-movies-dataset

https://grouplens.org/datasets/movielens/100k/

https://medium.com/swlh/getting-started-with-recommender-systems-5ad8846c280b

https://towardsdatascience.com/intro-to-recommender-system-collaborative-filtering-64a238194a26

https://www.coursera.org/learn/machine-learning