Movies Recommendation Engine

SHUBHAM PATIL
ECKOVATION
PYTHON PROGRAMMING

INTRODUCTION

Intro

A recommendation system is a type of information filtering system which attempts to predict the preferences of a user, and make suggestions based on these preferences.

E.g: People who buy the Apple Macbook also buy a USB-C to USB Adapter, they can recommend the Adapter to an user who a Macbook to his cart.

Types of Recommendation

Content Based Recommendation

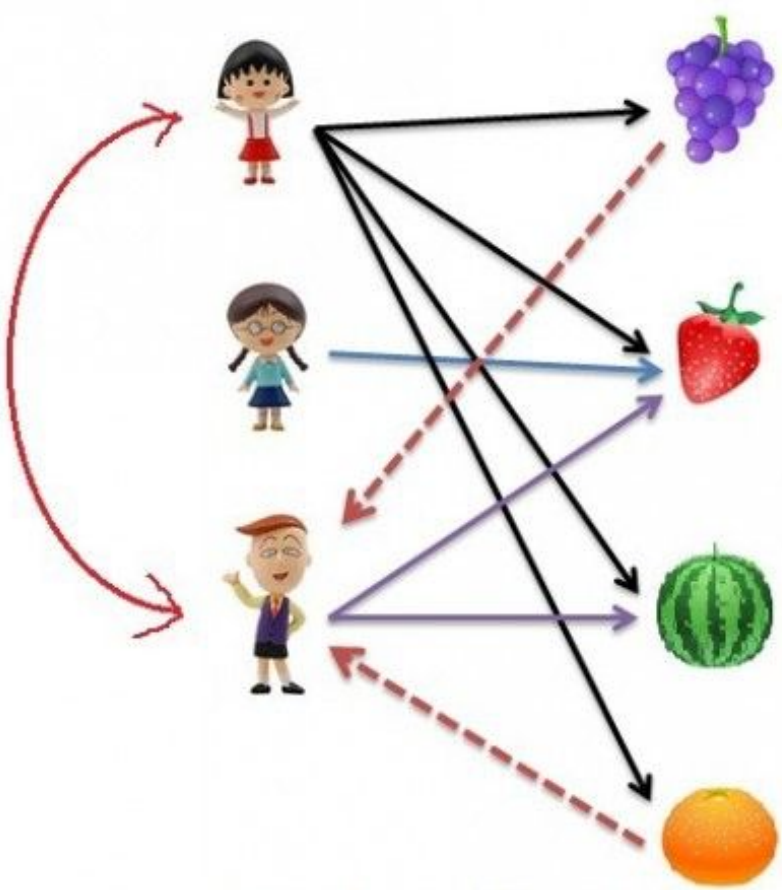
A content based recommender works with data that the user provides.

Based on that data, a user profile is generated, which is then used to make suggestions to the user.

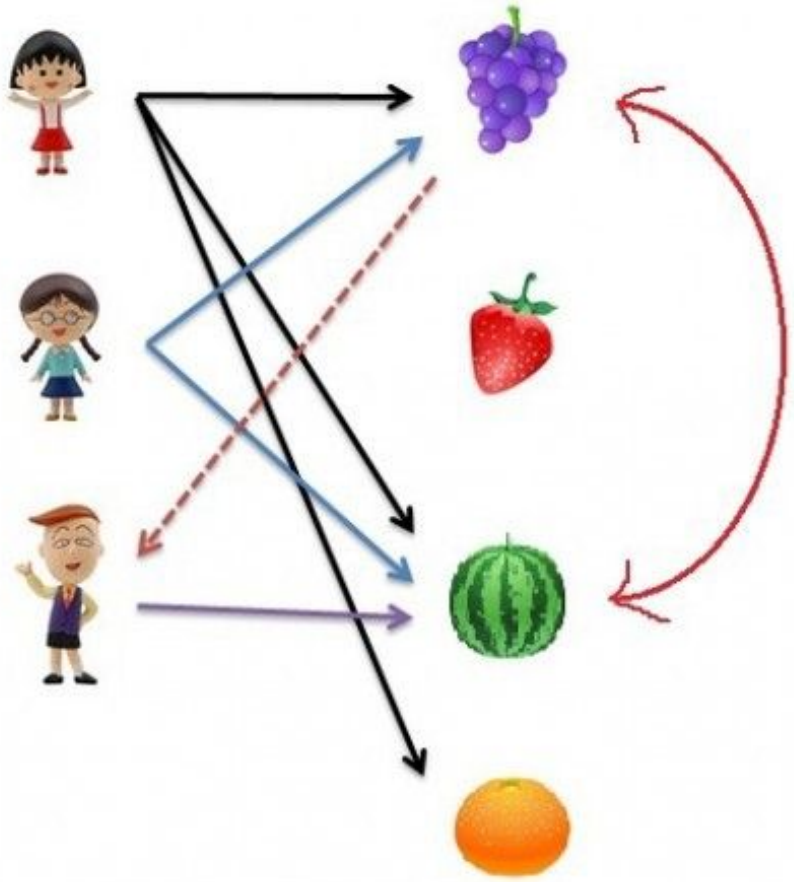
Collaborative Filtering

The basic idea of CF-based algorithms is to provide item recommendations or predictions based on the opinions of other like-minded users.

The opinions of users can be obtained explicitly from the users or by using some implicit measures



User-based filtering



Item-based filtering

Steps for Program

1. Import Dataset
2. Edit using python
   a. Clear unwanted data
   b. Merge useful columns
3. Display format of Dataset
4. Import Scikit Learn
5. Fit model using KNN
6. Use Random to choose a Movie
7. List of Recommendation are displayed beside.

Out[15]:

userId	1	2	3	4	5	6	7	8	9	10	...	601	602	603	604	605	606	607	608	609	610
title																					
10 Things I Hate About You (1999)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	...	0.000	0.000	3.000	0.000	5.000	0.000	0.000	0.000	0.000	0.000
12 Angry Men (1957)	0.000	0.000	0.000	5.000	0.000	0.000	0.000	0.000	0.000	0.000	...	5.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
2001: A Space Odyssey (1968)	0.000	0.000	0.000	0.000	0.000	0.000	4.000	0.000	0.000	0.000	...	0.000	0.000	5.000	0.000	0.000	5.000	0.000	3.000	0.000	4.500
28 Days Later (2002)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	...	0.000	0.000	0.000	0.000	0.000	0.000	0.000	3.500	0.000	5.000
300 (2007)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	3.000	...	0.000	0.000	0.000	0.000	3.000	0.000	0.000	5.000	0.000	4.000

5 rows × 606 columns

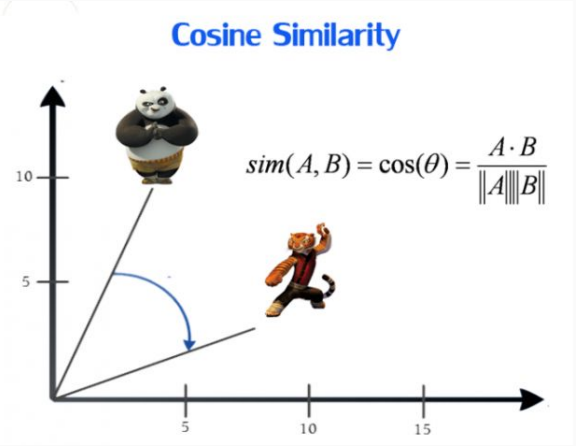
In [16]:
```
for i in range(0, len(distances.flatten())):
    if i == 0:
        print('Recommendations for {0}:\n'.format(movie_features_df.index[query_index]))
    else:
        print('{0}: {1}, with distance of {2}:'.format(i, movie_features_df.index[indices.flatten()[i]], distances.flatten()[i]))
```

```
Recommendations for Million Dollar Baby (2004):

1: Mystic River (2003), with distance of 0.5117906332015991:
2: Kill Bill: Vol. 1 (2003), with distance of 0.5247379541397095:
3: Crash (2004), with distance of 0.5259743928909302:
4: Star Wars: Episode III - Revenge of the Sith (2005), with distance of 0.526322066783905:
5: Sin City (2005), with distance of 0.5547516345977783:
```

```
Recommendations for Million Dollar Baby (2004):

1: Mystic River (2003), with distance of 0.5117906332015991:
2: Kill Bill: Vol. 1 (2003), with distance of 0.5247379541397095:
3: Crash (2004), with distance of 0.5259743928909302:
4: Star Wars: Episode III - Revenge of the Sith (2005), with distance of 0.526322066783905:
5: Sin City (2005), with distance of 0.5547516345977783:
```

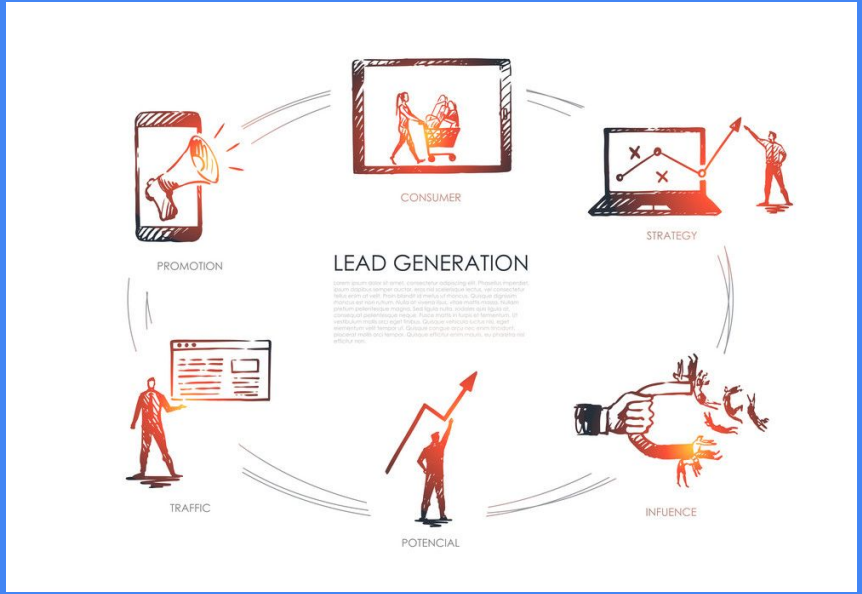
Similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them. Above, the **least distance** proves how similar is the Recommended Movie.

$$\text{Distance} \propto \frac{1}{\text{Similarity between recommended movie}}$$



Advantages of content-based filtering

- They are capable of recommending unrated items.
- We can easily explain the working of the recommender system by listing the Content features of an item.
- Content-based recommender systems use only the rating of the concerned user,and not any other user of the system.



RESULTS

A hybrid approach is taken between context based filtering and collaborative filtering to implement the system.

This approach overcomes drawbacks of each individual algorithm and improves the performance of the system.

Here, for movie with ID 270, we get an estimated similarity of 0.511.

Similarity and Classification are used for better recommendation and increasing accuracy and precision.

Thank You!

Shubham Patil
Eckovation
Python Programming