International Journal of Computer Architecture and Mobility (ISSN 2319-9229)
Volume 7 -Issue 10, October 2019

Movie Recommendation System

Abhishek Singh
Shri Vaishnav Vidyapeeth Vishwavidyalaya
Indore,India
abhiasingh1004@gmail.com

Abstract—**In this paper i propose a movie recommendation system that uses both content based filtering and collaborative filtering and here i build a website that uses this recommendation system. This web site also shows breaking news about the Bollywood and Hollywood, it also has music and book recommendation pages. Other features of our website are it show trailers of the new movies, upcoming movies and web series.**

I. INTRODUCTION

Recommendation systems are becoming more and more popular, with every major website incorporating them into their system, they are at an all time high. With this in mind, giving users quick but accurate recommendations, is more important than ever. This document describes a movie recommender system that is designed to provide users with recommendations to help them decide what movie(s) to watch next. It will look at previous and existing solutions and assess each system, and use this information in order to formulate a new app to help users get movie recommendations. These recommendations will be achieved using a sophisticated maths algorithm.

It is hard to imagine the existence of e-commerce websites or any other popular entertainment and social media sites that do not utilize recommender systems nowadays. Recommender system is a technology to filter information in a website or a system in order to predict the rating or preference of a product for users. It has made a massive change in the way people interact with websites. In e-commerce websites, for example, users are easily guided to products they like according to their preference and taste based on their past shopping information. Another example can be taken from social networking sites, where users are suggested to connect with other users they may know based on their mutual interest, friends, or occupation. This improvement has certainly created a more attractive online experience for users by giving an ease of access through recommendations from the system.

Recommender system can be of two types:-

1. Content based recommender system.
2. Collaborative recommendation system.

1.1 Content-based filtering

This approach requires a good amount of information of items' own features, rather than using users' interactions and feedbacks. For example, it can be movie attributes such as genre, year, director, actor etc., or textual content of articles that can be extracted by applying Natural Language Processing.

Advantages [3]:-

- The model doesn't need any data about other users, since the recommendations are specific to this user. This makes it easier to scale to a large number of users.
- The model can capture the specific interests of a user, and can recommend niche items that very few other users are interested in.

Disadvantages [3]:-

- Since the feature representation of the items are hand-engineered to some extent, this technique requires a lot of domain knowledge. Therefore, the model can only be as good as the hand-engineered features.
- The model can only make recommendations based on existing interests of the user. In other words, the model has limited ability to expand on the users' existing interests.

1.2 Collaborative filtering

On the other hand, collaborative filtering doesn't need anything else except users' historical preference on a set of items. Because it's based on historical data, the core assumption here is that the users who have agreed in the past tend to also agree in the future.

Advantages [4]:-

- **No domain knowledge necessary**

We don't need domain knowledge because the embeddings are automatically learned.

- **Serendipity**

The model can help users discover new interests. In isolation, the ML system may not know the user is interested in a given item, but the model might still recommend it because similar users are interested in that item.

- **Great starting point**

To some extent, the system needs only the feedback matrix to train a matrix factorization model. In particular, the system doesn't need contextual features. In practice, this can be used as one of multiple candidate generators.

Disadvantages [4]:-

1. **Cannot handle fresh items**

The prediction of the model for a given (user, item) pair is the dot product of the corresponding embeddings. So, if an item is not seen during training, the system can't create an embedding for it and can't query the model with this item. This issue is often called the **cold-start problem**. However, the following techniques can address the cold-start problem to some extent:

- **Projection in WALS.** Given a new item i0 not seen in training, if the system has a few interactions with users, then the system can easily compute an embedding ui0 for this item without having to retrain the whole model. The system simply has to solve the following equation or the weighted version:

$$\min_{u_{i_0} \in \mathbb{R}^d} \| A_{i_0} - u_{i_0} V^T \|$$

The preceding equation corresponds to one iteration in WALS: the user embeddings are kept fixed, and the system solves for the embedding of item i0. The same can be done for a new user.

- **Heuristics to generate embeddings of fresh items.** If the system does not have interactions, the system can approximate its embedding by averaging the embeddings of items from the same category, from the same up loader (on YouTube), and so on.

2. **Hard to include side features for query/item**

Side features are any features beyond the query or item ID. For movie recommendations, the side features might include country or age. Including available side features improves the quality of the model. Although it may not be easy to include side features in WALS, a generalization of WALS makes this possible.

To generalize WALS, **augment the input matrix with features** by defining a block matrix $\bar{A}$, where:

- Block (0, 0) is the original feedback matrix A.
- Block (0, 1) is a multi-hot encoding of the user features.
- Block (1, 0) is a multi-hot encoding of the item features.

2.Literature Survey

2.1 KNN Algorithm

The KNN algorithm assumes that similar things exist close proximity. In other words, similar things are near to each other. KNN captures the idea of similarity (sometimes called distance, proximity, or closeness) with some mathematics we might have learned in our childhood— calculating the distance between points on a graph. The idea of KNN Algorithm is that if k nearest neighbor belong to certain category than item belong to that category.

3.Similarity measures

The similarity measure is the measure of how much alike two data objects are. Similarity measure in a data mining context is a distance with dimensions representing features of the objects. If this distance is small, it will be the high degree of similarity where large distance will be the low degree of similarity.

The similarity is subjective and is highly dependent on the domain and application. For example, two fruits are similar because of color or size or taste. Care should be taken when calculating distance across dimensions/features that are unrelated. The relative values of each element must be normalized, or one feature could end up dominating the distance calculation. Similarity are measured in the range 0 to 1. (1)

A. Euclidean distance:

Euclidean distance is the most common use of distance. In most cases when people said about distance, they will refer to Euclidean distance. Euclidean distance is also known as simply distance. When data is dense or continuous, this is the best proximity measure.

The Euclidean distance between two points is the length of the path connecting them. The Pythagorean theorem gives this distance between two points.

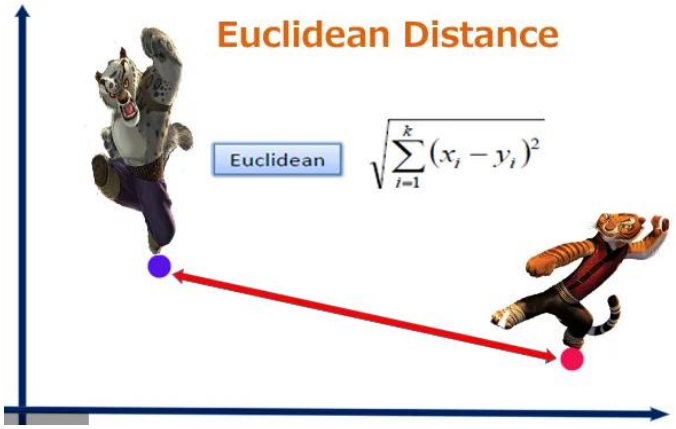

Figure 1.1 Euclidean Distance [1]

B. Manhattan distance:

Manhattan distance is a metric in which the distance between two points is the sum of the absolute differences of their Cartesian coordinates. In a simple way of saying it is the total suzm of the difference between the x-coordinates and y-coordinates.

Suppose we have two points A and B if we want to find the Manhattan distance between them, just we have, to sum up, the absolute x-axis and y – axis variation means we have to find how these two points A and B are varying in X-axis and Y- axis. In a more mathematical way of saying Manhattan distance between two points measured along axes at right angles.

In a plane with p1 at (x1, y1) and p2 at (x2, y2).

Manhattan distance = |x1 – x2| + |y1 – y2|

This Manhattan distance metric is also known as Manhattan length, rectilinear distance, L1 distance or L1 norm, city block distance, Minkowski's L1 distance, taxi-cab metric, or city block distance.

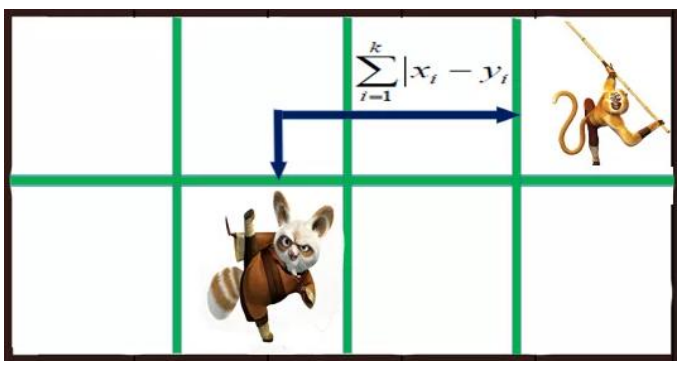

Figure 1.2 Minkowski distance [1]

C. Cosine similarity:

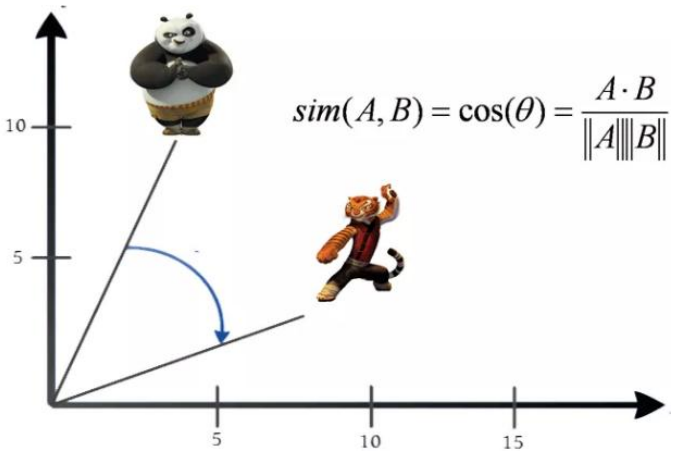

Figure 1.3 Cosine similarity [1]

Cosine similarity metric finds the normalized dot product of the two attributes. By determining the cosine similarity, we would effectively try to find the cosine of the angle between the two objects. The cosine of 0° is 1, and it is less than 1 for any other angle.

It is thus a judgment of orientation and not magnitude: two vectors with the same orientation have a cosine similarity of 1, two vectors at 90° have a similarity of 0, and two vectors diametrically opposed have a similarity of -1, independent of their magnitude.

Cosine similarity is particularly used in positive space, where the outcome is neatly bounded in [0,1]. One of the reasons for the popularity of cosine similarity is that it is very efficient to evaluate, especially for sparse vectors.

4. DATA DESCRIPTION

In the proposed paper we are using both collaborative filtering and content based filtering for recommending personalized results to an individual and collaborative filtering for recommending movies and music based on other users.
For recommending personalized results we are using Movielens dataset .Attributes used for this are

- Genre
- Release Year
- Rating
- Actors

For predicting based on collaborative filtering we are using again MovieLens dataset.
MovieLens is a web-based recommender system and virtual community that recommends movies for its users to watch, based on their film preferences using collaborative filtering of members' movie ratings and movie reviews. It contains about 11 million ratings for about 8500 movies.

5. SIMULATION OF APPLICATION

This application is created using Flask which is python's micro framework for web development. The home route

mainly contains all the movie by genres clicking which user can view the movies. It contains the Go button clicking which user can comes to Rating route on which user can rate the movies.
Here user is allowed to rate the movies based on Genres.
to take the ratings given by user we are using List called preferences[] in which we are inserting rating for each geners given by users. These ratings are in specific order like:-

[Action,Adventure,sci-fi,etc] : [1,0,2,etc]

```
[8570 rows x 18 columns]
[2, 0, 1, 4, 0, 2, 0, 0, 0, 0, 0, 0, 3, 0, 0, 4, 0, 0]
```

After doing all the preprocessing we got the List called categories[] which is 0-1 matrix which contains movies and their gieners.

```
      Action  Adventure  Animation  Children  ...  Sci-Fi  Thriller  War  Wester
n
0          0          1          1         1  ...       0         0    0
0
1          0          1          0         1  ...       0         0    0
0
2          0          0          0         0  ...       0         0    0
0
3          0          0          0         0  ...       0         0    0
0
4          0          0          0         0  ...       0         0    0
0
...      ...        ...        ...       ...  ...     ...       ...  ...      ..
.
8565       1          0          0         0  ...       0         0    0
0
8566       0          0          0         0  ...       0         0    0
0
8567       1          0          0         0  ...       0         1    0
0
8568       0          0          0         0  ...       0         0    0
0
8569       0          0          0         0  ...       0         0    1
0

[8570 rows x 18 columns]
127.0.0.1 - - [26/Sep/2019 22:53:09] "POST /recommendation HTTP/1.1" 200 -
```

From this we can calculate rating of each movie based on user preference.We can do this by using Simple Dot product.So here we calculate the dot product of categories and preference to calculate the rating of each movie.This can be written as:

movies[score]=dot_product(categories,preference)

By doing this we get the rating of each movie as:-

```
0          5
1          4
2          0
3          0
4          0
          ..
8565       2
8566       0
8567       6
8568       0
8569       0
Name: score, Length: 8570, dtype: int64
```

This is main funda i used for content based filtering.

When user opens the application he needs to register to the movie site. After registration he need to Login to the system. After he does so he can see:

- Navbar
- Search bar
- Side bar
- Home screen

On Navbar he can see all the other options like web series, music and games and etc. In search bar he can search the movies. On sidebar he can see the latest movie related news the news can be of Hollywood, Bollywood, and webseries and about actors. The rest of the home screen contains all the movies of different genres.

For content based filtering we taking response from user by directing user to rating page in which we ask view of user about different types of movies.

Than we are using those ratings to recommend movies to user liked by user.

6. WEB SCRAPPING

The major part of the project is covered by web scrapping. It is a technique used to extract data from the internet. This data is enclosed under html tags. So to access this data we need to use html tags.

Python provides following features for web scraping like[5]:-

- Ease of Use: Python is simple to code. You do not have to add semi-colons ";" or curly-braces "{}" anywhere. This makes it less messy and easy to use.
- Large Collection of Libraries: Python has a huge collection of libraries such as Numpy, Matlplotlib, Pandas etc., which provides methods and services for various purposes. Hence, it is suitable for web scraping and for further manipulation of extracted data.
- Dynamically typed: In Python, you don't have to define datatypes for variables, you can directly use the variables wherever required. This saves time and makes your job faster.
- Easily Understandable Syntax: Python syntax is easily understandable mainly because reading a Python code is very similar to reading a statement in English. It is expressive and easily readable, and the indentation used in Python also helps the user to differentiate between different scope/blocks in the code.
- Small code, large task: Web scraping is used to save time. But what's the use if you spend more time writing the code? Well, you don't have to. In Python, you can write small codes to do large tasks. Hence, you save time even while writing the code.
- Community: What if you get stuck while writing the code? You don't have to worry. Python community has one of the biggest and most active communities, where you can seek help from.

Steps for web scrapping:-

1. Find the URL that you want to scrape
2. Inspecting the Page
3. Find the data you want to extract
4. Write the code
5. Run the code and extract the data
6. Store the data in the required format

Libraries used for web-scrapping:-

- Selenium: Selenium is a web testing library. It is used to automate browser activities.
- BeautifulSoup: Beautiful Soup is a Python package for parsing HTML and XML documents. It creates parse trees that is helpful to extract the data easily.

- Pandas: Pandas is a library used for data manipulation and analysis. It is used to extract the data and store it in the desired format.

8. CONCLUSION

So in the following paper I have developed a movie recommendation website that is capable of predicting movies based on user's preference.This site is not only a movie recommendation system but it also has the functionalities of games,news,watching trailers of new movies.This is made possible with the help of web scraping.User can view Bollywood and hollywood breaking ,user can view the latest movie arrival,user can play online games,user can watch online TV series on my site.In conclusion i can say that this site is all in one site for all the related works.

Although the work is not completed as this site is under development for Collaborative filtering.But for content based filtering ,here i am using sime Dot product for providing personalized recommendations instead of using complex algorithm.Here i am predicting the rating of all the movies according to user's likes and dislikes.

7. REFERENCES

[1] Dataaspirant, FIVE MOST POPULAR SIMILARITY MEASURES, Dewas, 2019.

[2] S. Polamuri, "FIVE MOST POPULAR SIMILARITY MEASURES IMPLEMENTATION IN PYTHON," Dataaspirant, [Online]. Available: https://dataaspirant.com/2015/04/11/five-most-popular-similarity-measures-implementation-in-python/. [Accessed 29 9 2019].

[3] developers.google.com, "Content-based Filtering Advantages & Disadvantages," google, [Online]. Available: https://developers.google.com/machine-learning/recommendation/content-based/summary. [Accessed 30 9 2019].

[4] developers.google.com, "Collaborative Filtering Advantages & Disadvantages," Google, [Online]. Available: https://developers.google.com/machine-learning/recommendation/collaborative/summary. [Accessed 30 9 2019].

[5]. **edureka.** A Beginner's Guide to learn web scraping with python! *edureka.* [Online] edureka. [Cited: 3 10 2019.] https://www.edureka.co/blog/web-scraping-with-python/.