Movie Recommendation System

Anant Vaid
CSE
anantvaid4@gmail.com
People Education Society
University EC- Campus
Bengaluru, Karnataka

Shashank S Byakod
CSE
shashanksbyakod37@gmail.com
People Education Society
University EC- Campus
Bengaluru, Karnataka

Gourav Pujari
CSE
gouravpujari6@gmail.com
People Education Society
University EC- Campus
Bengaluru, Karnataka

Abstract*— Over the past years, the internet has broadened the horizon of various domains to interact and share meaningful information. As it is said that everything has its pros and cons therefore, along with the expansion of domain comes information overload and difficulty in extraction of data. To eradicate this problem the recommendation systems imparts very important role. It is used to enhance the experience by giving fast and coherent suggestions. This paper describes an approach which offers generalized recommendations to every user, based on genres, keywords, cast of the movie. Collaborative-Based filtering and Content Based filtering technique being implemented using few NLP approaches.

Keywords--- Recommendation System, Collaborative-based filtering(KNN, user based ,item based), Content-based filtering(Cosine Similarity)

I. INTRODUCTION

This is an Era of entertainment and business Movies are a source of economic growth and is a demanding area for entertainment all over the world. Everyone has a liking towards a particular genre, that varies from person to person. People love to ask others for movie opinions and watch movies of same kind if they like it. Movie Recommenders do the job of recommending people, a movie that follows their watch pattern and also help the company earn business. It is highly important for the company(Netflix, Amazon Prime Video, etc) to understand the user preference pattern, in order to keep them hooked to their platform and be able to make business and gain trust. Recommendation systems are primarily using three approaches. In content-based filtering, we do profiling based on what type of content any user is interested in and using the collected information, it recommends items. Another one is collaborative filtering, where we make clusters of similar users and use that information to make recommendations. Hybrid systems are the one which takes into account both above stated approaches to do with operational data more concisely.

II. Previous Work

- ***Review 1:***

Reviewed: Anant Vaid

A. Strategy Used

This paper [1] helps show light on the two most infamous filtering algorithms that are Collaborative and Content Filtering techniques. The former technique uses a supervised model that uses Bayesian approach that forms recommendations based on the previous evaluations of other movies the user has watched. Whereas, the latter technique uses unsupervised approach to cluster the users, using K-Mean clustering algorithm, based on the preference of different movie plots, genres, etc through their ratings. Both of the techniques make recommendations based on the Distance metric and cosine similarity between watchers and films (KNN). Content (Base) Filtering, as the name suggests uses K-Means Algorithm, to find resemblance in the movie keywords of plots.

The 1st way clusters movies based on the **TF-IDF** (i.e Term Frequency Inverse Term Frequency) weighting scheme (pre-processing step), placing weights on the terms of use of movie plots. Term frequency is given as number of repetitive terms in the document to the number of terms in the document. Inverse Document Frequency *is given as log(number of documents/documents containing the term)*. The overall is equal to TF * IDF.

B. Natural Language Processing

NLP techniques are involved in pre-processing the text data, like tokenisation, stop words removing, punctuations & number removing, POS tags, lower casing and stemming or lemmatisation. These are done because we use the movie plots made up of sentences rather than keywords. These pre-processed words are later passed through Tf-Idf vectorization to vectorize the text data. It gives weights according to the frequency of occurrence of the words in the documents.

The large collection of vectors are now reduced using PCA that reduces the dimensions by squashing correlated vectors as one. The similarity metrics that can be used to find best recommendations are Cosine similarity between vectors and Euclidean distance.

C. Predicting User Ratings

User-Choices can be simulated by M*N. M has the full number of movie-watchers and N has full number of movies.

We predict the user ratings user-item matrix, with each user being represented as a vector, and the components are ratings of films they rated. Similarly, movie is represented by vector containing evaluations by watchers. The agenda is to impute the values that are missing with the ratings that are generated by the model.

Review 2:

Reviewed: Shashank S Byakod

Movie Recommendation System using Cosine Similarity and KNN.

A. Overview

Here in this paper [2] discussing on three known recommendation system that are Content-Based Recommendation system, Collaborative-based Recommendation system and Hybrid-based Recommendation system. In Content Based Recommendation system, we will be profiling basis on what type of content any user is interested in and using the collected information. Another one is collaborative filtering, where we make clusters of similar users and use that information to make recommendations. And the latter approach Hybrid filtering are the one which takes into account both above stated approaches to deal with operational data more concisely. As already stated above that in Content Based filtering the items based on the liking of the user. And here the result obtained based on what the user has rated earlier. They have used Vector Space Model (VSM) for this approach. It derives the similarity of the item from its description and then talking on,

TF- IDF(i.e Term Frequency – Inverse Document frequency).

$$Tf(t) = \frac{frequency\ occurrence\ of\ term\ 't' \in document}{total\ number\ of\ terms\ in\ document\ 't'}$$

$$If(t) = log10 * \frac{total\ number\ of\ document}{number\ of\ documents\ containing\ term\ 't'}$$

There are some problems related to the recommendation system as follows,

• **Cold start problem:** When a user/client register for the first time,though he hasn't watched any movie yet. So then the recommender could not able to give recommendation for that particular user , because the recommendation system works on previous record of the user.

• **Data sparsity problem:** When the user has rated few items, then later recommendation system could not provide the accurate result due to lack of information from that particular user. Here it might fail in giving the desired output.

• **Scalability:**As data set increases it becomes difficult for the recommendation system to give accurate results based on varying genres of movies. If in this cases arises, then it can be solved by the dimensionality reduction technique (SVD).It could speed up the generation of the output and produces a efficient result.

• ***Synonym:***The recommendation system fails to understand the difference between two similar words, this might leads to undesired output.So this can be resolved by the SVD technique with Latent Semantic Indexing.

B. Collaborative filtering

In Collaborative filtering,the recommendation are based on the the ratings and the genres given other set of user , so this may not give acurate output.This is the drawback of Collaborative filtering. In this case, we might look for another approach that is Content Based filtering.Here in this method it uses the rating and genres given by the same user itself. It gives recommendations based on the movies watched by the user earlier. In former approach it lacks the accuracy of true recommendations due to their dependency on other users.

C . Cosine Similarity

Here is the formulae to find the similarity between the two vectors(here movies) as follows:

$$CosSim(x, y) = \frac{\sum_i x_i y_i}{\sqrt{\sum x_i^2}\sqrt{\sum y_i^2}}$$

The angle theta between two movies will determine the similarity between the two movies. The above CosSim(x,y) values ranges from 0 to 1 .If the value is nearer to 1 then it is similar ,if incase it is near to 0 then it is least similar.The movie would be recommended if it is close to 1. Along with the cosine similarity , by using KNN functionalty that used to find the nearest neighbour which will be recommended to the user.

Review 3:

Reviewed: Gourav Pujari

This paper [3] was designed to provide a simple and lower-cost movie recommendation system huge cultural json format, about movies, on the Web and analysed the advantasge of the system. As a result, we could knowing the potential of cultural metadata. The Dataset is also very similar for what we have chosen for our project analysis .

Usually recommendation systems are'nt that easy to compute and feasible, Here in this paper the motive is to build low cost recommendation system containing vast cultural metadata, about movies, on the Web and analyzed the strength of the system. Because of which we can learn and understand importance of cultural metadata. The Dataset is also very similar for what we have choosen for our project analysis.

There are many models for recommendation system, these methods are few among them, there are two main streams of

approaches to recommendation system research, **CBR (Content-Based Recommendation)** and **CF (Collaborative Filtering).**

- CBR or Content-Based Recommendation analyzes the contents of new items and recommends those which are the most similar to the items which the user has liked in past or has ordered or enquired about those.
- CBR is mainly used in IR (Information Retrieval) techniques and, therefore it easily handles textbased items like news books ,articles, URL, and many other journals and stuff.
- Oflate CBR also deals with many other kinds of media like music, pictures , drawings and web series, as text-based metadata for all of these media are created and provided widely, this feature makes CBR deal with variety and heterogeneous data.
- The best way of implementing CBR is to provide the most similar items to the target item tat user is interested in regardless of users individual preferences.

Uses: song albums recommended by yahoo music.
Never the less, CBR has several shortcomings , ofcourse every model has one or the other shortcomings. The techniques in which you can understand or analyze the data are limited. Only a very shallow analysis of certain kinds of content can be supplied.

III. Proposed Solution

We will be using KNN and Cosine Similarity:

- **Recommendation using Cosine Similarity:**

It is Content-based filtering technique, wherein we would be finding the similarities between the two vectors (two different rows from the dataset) by applying the cosine of two vectors. Here, as per our data, we use the cast, genre and the keywords (plot of the movie) and cascade them as a single text and use the infamous Tf - Idf transformation to convert the text to vector for further vector calculations.

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$

After getting the vectors, we will using the cosine-similarity technique between vectors to get the cosine-similarity matrix. This matrix is used to get the cosine similarity vector for the movie based on indexing, and we sort them in on-increasing manner to get the top ranked similar movies. This is just using cosine similarity and extracting the top similar movies, there is no train test split of dataset, as we cannot evaluate the results. This is purely suggestions that mostly show the movies that are close to the cast, genre and the keywords.
The technique used above is purely content based and does not consider or pool similar users based on the ratings provided by them. We solely use the textual data that we get from the cast of the movie, its genre and the keywords to suggest/ recommend a movie.
For further more recommendation, we will be making use of KNN algorithm (unsupervised learning).

- **Recommendation using KNN (Collaborative based filtering):**

KNN falls under the **supervised** algorithms. We make use of this approach and the data containing the users and their movie ratings. The estimated rating is basically a weighted mean of the ratings the user gave to familar items, weighted by the similarities.

$$\hat{r}_{ui} = \frac{\sum_{v \in N_i^k(u)} \text{sim}(u,v) \cdot r_{vi}}{\sum_{v \in N_i^k(u)} \text{sim}(u,v)}$$

$$\hat{r}_{ui} = \frac{\sum_{j \in N_u^k(i)} \text{sim}(i,j) \cdot r_{uj}}{\sum_{j \in N_u^k(i)} \text{sim}(i,j)}$$

Here, the prediction r_{ui} is weighted sum of ratings with the similarity. We see how r_{ui} , the rating user u would give to item i, is estimated in the predictions. The formulas *above* mostly use notations we have discussed in the previous section, a couple of new ones: σ_i is the standard deviation of item i, $N_u^k(i)$ is the maximum k items from the ones user u rated that are closest to item i.

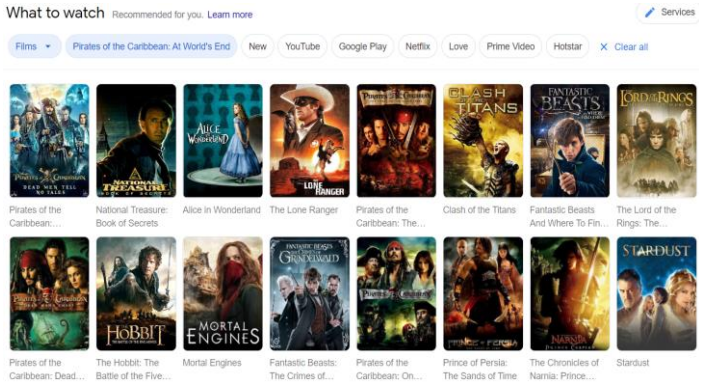
IV. Experimental Results

We tested out the two proposed solutions with our dataset and were able to get the recommendations accordingly.
Using **only Cosine similarity technique** which we pondered upon in the previous section, we provided the output which was the similarity in the text containing the cast, genre and the keywords. We tested out on few movies and got the respective results.

```
similarity: [[1.         0.0327055  0.021148   ... 0.01375698 0.0095136  0.        ]
 [0.0327055  1.         0.01943422 ... 0.01264215 0.00874265 0.        ]
 [0.021148   0.01943422 1.         ... 0.         0.05767289 0.00748532]
 ...
 [0.01375698 0.01264215 0.         ... 1.         0.00561601 0.        ]
 [0.0095136  0.00874265 0.05767289 ... 0.00561601 1.         0.01081919]
 [0.         0.         0.00748532 ... 0.         0.01081919 1.        ]]
```

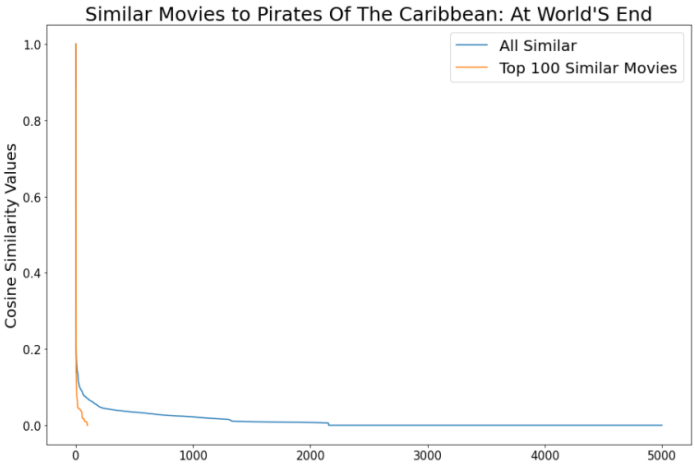
This was the cosine similarity matrix that we got from the data. Each row represents how close that movie is with respect to all other movies based on row indexing. This was our basis of recommendations. After taking the name of the movie from the user, we predicted the results of the most similar 10 movies for the respective movie.
Lets consider movie "pirates of the caribbean: at world's end":
And the recommender system output as follows:

```
The top 10 recommended movies based on your search are
                Pirates Of The Caribbean: The Curse Of The Black Pearl
                Pirates Of The Caribbean: Dead Man'S Chest
                The Lone Ranger
                Rango
                The Libertine
                Pirates Of The Caribbean: On Stranger Tides
                Permission
                The Lord Of The Rings: The Return Of The King
                Corpse Bride
                The Mexican
```



A look into the recommendation by Google shows that our recommender is at par with them, with most of the similar movies, just by extracting textual information.

The below line plot depicts the cosine similarity between a specific movie (pirates of the caribbean: at world's end) with the similar movies. The orange colored line refers to the most similar movies and the blue colored line refers to the other remaining movies.



Using KNN with Cosine similarity on user data and their ratings provided, we used the weighted average of ratings over the similarity of K nearest neighbors for the user. There are two strategies applied, one with user consideration and other with item consideration. We group the information either by assuming similar items based on the ratings, or the users based on their ratings.
On performing the cross-validation on this model, we have found that KNNBasic algorithm had test_rmse value of 0.958377.

```
Computing the msd similarity matrix...
Done computing similarity matrix.
Computing the msd similarity matrix...
Done computing similarity matrix.
Computing the msd similarity matrix...
Done computing similarity matrix.
```

Algorithm	test_rmse	fit_time	test_time
KNNBasic	0.958377	0.117922	2.542367

Using used based similarity, we would be finding the recommendation for **userId 6:**

```
0                                    Crumb (1994)
1                          Grifters, The (1990)
2                   Black Cauldron, The (1985)
3                                      Go (1999)
4     Star Wars: Episode I - The Phantom Menace (1999)
5         Austin Powers: The Spy Who Shagged Me (1999)
6                   Dirty Dozen, The (1967)
7                            Magnolia (1999)
8                          Wayne's World (1992)
9                          Thirteen Days (2000)
10                 Louis C.K.: Hilarious (2010)
Name: title, dtype: object
```

Using the item based similarity, we find the recommendation for the random userId 6:

```
0                          Prefontaine (1997)
1                        Excess Baggage (1997)
2       Twin Peaks: Fire Walk with Me (1992)
3                  Where the Heart Is (2000)
4                           Black Rain (1989)
5                      Waterdance, The (1992)
6     Ferngully: The Last Rainforest (1992)
7                           American Me (1992)
8                           Open Range (2003)
9                        Running Scared (2006)
10                          Half Nelson (2006)
Name: title, dtype: object
```

The item-based similarity shows the similar rated movies for that user, whereas the user-based similarity recommends the movies that the similar user (based on his/her rating patterns) has rated.

V. Conclusion

Recommendation systems are proves be an essential tool for businesses in order to get more users hooked. Future use of Sentiment Analysis can be considered for developing of more precise recommender system. These types of methods are commonly used in online streaming services (Amazon Prime Video, Netflix, Hotstar, etc). In this paper we have classify various approaches of recommender system that are based on Content based and Collaborative based filtering.
We can deploy the solution to either web/app platforms with good UIs to help users stream and enjoy movies on a hectic day.
We as a team of three collaborated equally and were able to do good team work. We learnt a lot about the techniques used in recommendation systems and were exceptionally joyed with the outcomes of it. We learnt the skills of good team work, communication and working in a scenario where we had to work remotely. Gourav Pujari took the job of preprocessing the datasets and was able to do it incredibly well. Shashank and Anant worked on building the models that did the recommendation of the movies based on cosine similarity and recommendation of the movies based on KNN strategy.

Here are the few sources that we referred while modelling recommender system.
https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm
https://en.wikipedia.org/wiki/Cosine_similarity
https://surprise.readthedocs.io/en/stable/knn_inspired.html

VI. References

1. Iliopoulou, K., Kanavos, A., Ilias, A., Makris, C., & Vonitsanos, G. (2020). Improving Movie Recommendation Systems Filtering by Exploiting User-Based Reviews and Movie Synopses. IFIP Advances in Information and Communication Technology (Vol. 585 IFIP). Springer International Publishing. https://doi.org/10.1007/978-3-030-49190-1_17

2. Singh, R. H., Maurya, S., Tripathi, T., Narula, T., & Srivastav, G. (2020). Movie Recommendation System using Cosine Similarity and KNN. International Journal of Engineering and Advanced Technology, 9(5), 556–559. https://doi.org/10.35940/ijeat.E9666.069520

3. Shinhyun Ahn & Chung-Kon Shi, G.(2009). Exploring Movie Recommendation System Using Cultural Metadata. Z. Pan et al. (Eds.): Transactions on Edutainment II, LNCS 5660, pp. 119–134, 2009. Springer International Publishing. https://link.springer.com/chapter/10.1007/978-3-642-03270-7_9