Movie Recommendation System Using Semi-Supervised Learning

*Note: Sub-titles are not captured in Xplore and should not be used

Ms. Sushmita Roy
Post-Graduation Student
Department of Information Technology (IT)
Thakur College of Science and Commerce (TCSC)
Mumbai, India
sushmitaroy272010@gmail.com

Prof. (Mr.) Mahendra Sharma
Professor
Department of Information Technology (IT)
Thakur College of Science and Comerce (TCSC)
Mumbai, India
smahendra@live.com

Dr. (Mr.) Santosh Kumar Singh
Head of The Department (HOD)
Department of Information Technology (IT)
Thakur College of Science and Commerce (TCSC)
Mumbai, India
sksingh14@gmail.com

***Abstract*—In the current digital era, with the huge amount of technologies and tremendous amount of information being available at disposal over the Internet, a huge amount of data is made available to users. This results in a condition known as "information overload". Due to these, it is difficult for a person to search and access information for taking decisions to arrive at an effective conclusion. To perorate this nut, there are filtering systems for information, known as the recommendation system or recommendation engine, considered here in paper, that help a person in identifying significant and possible services or products of interest based on the preferences given by him/her. This results in searching through lots of results to find the one that the user actually needs. This can be in cases like searching for books, music, videos, job postings, etc. A recommendation system is hence needed to help recommend items to users which are more relevant and accurate and fulfils the user's needs and requirements. A movie recommendation system is used to recommend movies which match the user taste's and preferences. Several approaches exist to implement this system – popularity-based recommendation system, simple recommendation system, content-based filtering, collaborative-based filtering, metadata-based filtering, demographic-based filtering approach. In this paper we use the following approaches – simple recommendation system, content-based filtering approach, collaborative-based filtering approach.**

Keywords*—recommendation system, recommendation engine, semi-supervised learning, simple recommendation system, content-based filtering approach, collaborative filtering approach

I. Introduction

Have you ever thought of – which algorithm Google might be using for maximizing its revenue through target ads or what about those e-commerce websites which advocate through choices like – people who bought Product X also bought Product Y or how does Facebook automatically suggest people whom we could be friends with or how does Facebook suggest us to tag friends in the pictures which hav not been tagged?

The answer to all these is through Recommendation Engines. With the tremendous amount of information being available on the Internet today and the increase in the number of users, today it is of growing importance for companies to locate and provide them with the related piece of information according to user's preferences.

The power of data and the boost in the sales by the e-commerce and retail companies are being leveraged by implementing recommendation engines in their website. In brief, the goal of these systems is to predict the tastes of the users and recommend items more likely to them.

II. What Is Recommendation System Or Recommendation Engine?

Recommendation engines, popularly known as recommendation systems are simple algorithms that provide the most correct and accurate items to the users by filtering the items from a huge pool of information. Recommendation systems work by discovering data patterns in the dataset by learning from consumer choices and preferences and then predicts the outcome that best relates to the user's interests.

A typical business scenario for implementing recommendation system – consider a following situation from an e-commerce website which sells smart phones. With the increasing number of mobile phone users or customers each day, the main task of the website is to display the best choices of smart phones to the users keeping in mind their tastes, choices and preferences. This is where the recommendation engine is actually required for use.

III. Motivation

Businesses these days implement recommendation systems in their websites to study the tastes and preferences of their customers. This enables them to attain their business objectives. Some reasons include:

- increase in traffic to website
- right elaboration of marketing policy to meet the customers' choices
- promotion of a product

IV. Recommendation Systems – The Need Of The Commerce Era

Is it worthwhile to implement a recommendation engine? If one is running a business successfully, one could probably survive the market without a recommendation system. However, to leverage the power of data and to give a tremendous increase to the boost in the sales and increase in

978-1-7281-3694-3/19/$31.00 ©2019 IEEE

the earnings, give the user a better experience, one should seriously consider implementing a recommender system.

A good way to look into this is how companies have successfully paved a long way by implementing recommendation systems: -

A. Amazon

According to McKinsey, 35% of the purchases on Amazon are a result of the implementation of their recommender systems.

B. Alibaba

According to Alizila, during the global shopping festival in China on 11-Nov-2016, Alibaba achieved a significant growth of up to 20% using personalized landing pages.

C. YouTube

Recommendations are the reason why 70% of the people spend their time watching videos and listening to audios on YouTube.

D. Netflix

According to McKinsey, 75% of the people watching Netflix come from recommendations.

E. Employee Recommender System - Nextflix

According to a paper written by an executive, implementing an employee recommender system saves around $1 billion each year for Netflix.

With the growing information space, recommendation engines have thus become the need of the hour.

V. Objectives For Implementation Of Recommendation Engine

There are varied reasons for why businesses use recommendation systems. The objectives for implementing a recommendation system include:

- Increase in number of items consumed by the consumer.
- Eases the user to find and select products or items essential to them which could otherwise be difficult to search.
- Better understanding of user's choices, tastes and preferences.
- Increased customer satisfaction.

VI. Tasks Of A Recommender System

Some of the tasks of a recommender system include the following: -

- Find some good items needed for user that best fits the user's needs.
- Recommend a sequence of products to customers.
- Browsing through product catalogue – Some users simply browse the website to view the products without any actual purpose. In such cases, the task of the recommender systems is to help the users to browse items within the search scope of the user.
- Helping others – Certain users may be interested to drop a comment or review or rating of the item or product which they have used as they belief in benefitting the community.
- Influencing people – Certain users could be influencing others, convincing others to buy products from their preferred recommended websites.

VII. Applications Of Recommendation Systems

The most common applications of recommendation systems include the following: -

- Entertainment Industry – providing recommendations to customers for movies, music, etc.
- E-commerce websites – providing recommendations to customers of products to buy books, images, research articles, computers, laptops, televisions, etc.
- Content – providing recommendations to consumers for personalized newspapers, Web pages, documents, e-learning applications and e-mail filters.
- Services – providing recommendations to users of travel services, houses for rent, experts for consultation, match-making services, etc.

VIII. Research Methodology

A. Research Goals

The goal of this Movie Recommendation system project is to recommend movies to users. Since we are implementing different approaches, these approaches will provide different recommendations according to the parameter used for calculating the recommendations.

B. Source of Data Collection

Data needed for recommender systems are often achieved from user ratings after watching a movie or listening to a song, implied search engine queries, browsing history or purchase histories of users logged-in or from other sources of information and knowledge.

Secondary data was collected for conducting this study. The dataset used here consists of data for the movies released on or before July 2017 from the Movies Dataset. This file consists of metadata for all the thousands of films listed in the Movie Lens data set.

C. Approach

- Popularity-based Recommender System: -
- Simple Recommender System: - Here, we sort the movies based on the popularity of each movie rating and popularity and then display the top movies from the list. Additionally, we can pass as argument the genre of the movie and then get the top movies based on a particular genre.
- Content-based filtering approach: - The recommender system built using the above two approaches has severe limitations. First, it gives the same

recommendations to everyone regardless of the user's personal tastes or opinions. If a person who loves action movies and hates romantic movies looks at our charts, he/she probably wouldn't like most of the movies. If he/she looks at our charts be genre, then probably he still wouldn't be getting the best recommendations.

To personalise our recommendations more, build an engine that computes similarity between movies based on some metrics and then recommend movies that are more to a particular movie. We use the cosine similarity to find the similarities between movies.

Content-based recommenders can be built using different ways. Two of the ways used: -

- Overviews and Taglines
- Cast, Crew, Keywords and Genre

IX. Implementation Framework

The proposed system is developed using the following: -

A. *Technology - Django*

Django is a free and open sourced web application development framework which has been written in Python. Using the web application framework, makes development faster and easier. It follows the Model-View-Template pattern of architecture.

The following figure demonstrates how each of the components of the MVT pattern interact with each other to serve the user's request –

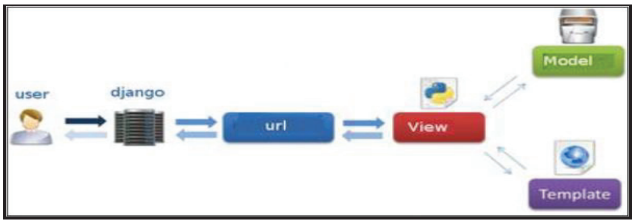


Fig. 1. MVT architecture of Django

The developer simply provides the model, the view and the template. It then just maps it to an URL. Django serves the web page to the user.

B. *Programming Paradigm - Python*

Python is a programming paradigm used to create web applications on the server. Soto give the application a graphical interface, we need to use Django framework and the backend tasks can be handled using Python.

C. *Methodology used – Machine Learning*

Machine Learning is an application of Artificial Intelligence (AI) which provide systems the ability to automatically learn.

Some methods of machine learning include: -

- Supervised Learning: - Supervised learning can only apply to the new data what it had learnt in the past using labelled example to predict the events.
- Unsupervised Learning: - Contrary to supervised learning, unsupervised learning algorithms are only used when the training data is neither labelled nor classified.
- Semi-supervised Learning: - Semi-supervised learning falls somewhere between the supervised learning and unsupervised learning. It uses both the labelled as well as the unlabelled data for training the dataset. The dataset under consideration uses semi-supervised learning mechanism.

X. Results

A. *Providing Recommendations: -*

a) Popularity-based Recommender System: -

Aaa.

b) Simple Recommender System: -

The Simple Recommender system offers generic recommendations to users based on movie popularity and sometimes genre. The basic idea for implementing such an approach is that movies that are more popular and more critically acclaimed will have higher chances of being liked on an average by the audience.

IMDB's weighted rating formula has been used to construct the chart. Mathematically it can be represented as follows using Equation (1):

$$\boxed{\text{Weighted Rating (WR)} = \left(\frac{v}{v+m}\ X\ R\right) + \left(\frac{m}{v+m}\ X\ C\right)}$$

where,

v - number of votes of a movie
m - minimum number of votes required for a movie to be featured in the top list charts
R - average rating of the movie
C - mean vote across the whole dataset

Results display recommendations based on weighted rating are shown in Fig. 2 below.

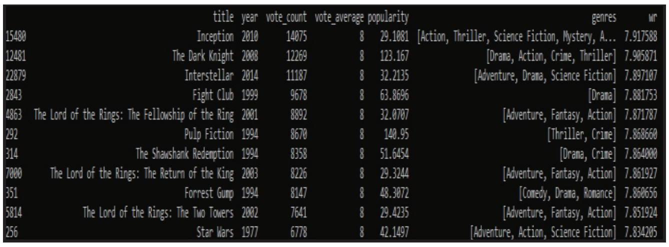
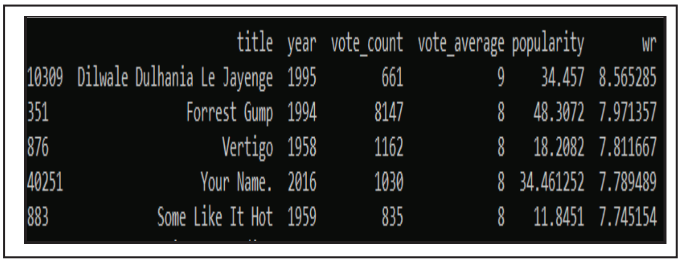

```
                                              title  year  vote_count  vote_average popularity                                          genres        wr
15480                                     Inception  2010       14075             8    29.1081  [Action, Thriller, Science Fiction, Mystery, A...  7.917588
12481                               The Dark Knight  2008       12269             8    123.167                  [Drama, Action, Crime, Thriller]  7.905871
22879                                  Interstellar  2014       11187             8    32.2135               [Adventure, Drama, Science Fiction]  7.897107
2843                                     Fight Club  1999        9678             8    63.8696                                           [Drama]  7.881753
4863  The Lord of the Rings: The Fellowship of the Ring  2001     8892             8    32.0707                     [Adventure, Fantasy, Action]  7.871787
292                                    Pulp Fiction  1994        8670             8     140.95                                [Thriller, Crime]  7.868660
314                        The Shawshank Redemption  1994        8358             8    51.6454                                   [Drama, Crime]  7.864000
7000    The Lord of the Rings: The Return of the King  2003      8226             8    29.3244                     [Adventure, Fantasy, Action]  7.861927
351                                    Forrest Gump  1994        8147             8    48.3072                          [Comedy, Drama, Romance]  7.860656
5814          The Lord of the Rings: The Two Towers  2002        7641             8    29.4235                     [Adventure, Fantasy, Action]  7.851924
256                                       Star Wars  1977        6778             8    42.1497             [Adventure, Action, Science Fiction]  7.834205
```

Fig. 2. Top Recommendations for movies based on simple recommender approach – by weighted average.

Results display recommendations based on genre are shown in Fig. 3 below.

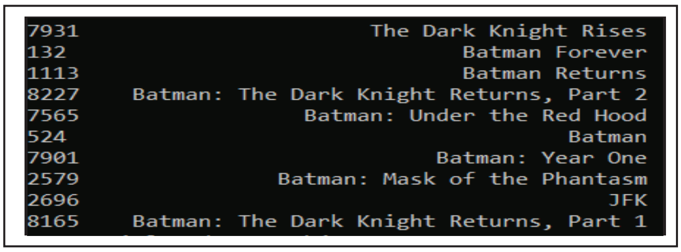


```
                        title  year  vote_count  vote_average  popularity        wr
10309  Dilwale Dulhania Le Jayenge  1995         661             9      34.457  8.565285
351                  Forrest Gump  1994        8147             8     48.3072  7.971357
876                       Vertigo  1958        1162             8     18.2082  7.811667
40251                  Your Name.  2016        1030             8   34.461252  7.789489
883                Some Like It Hot  1959         835             8     11.8451  7.745154
```

Fig. 3. Top Recommendations for movies based on simple recommender approach – by genre.

c) Content-based filtering approach: -

We try to fetch recommendations for movies which are similar to “The Dark Knight”. Result displays recommendations as shown in Fig. 4 below.

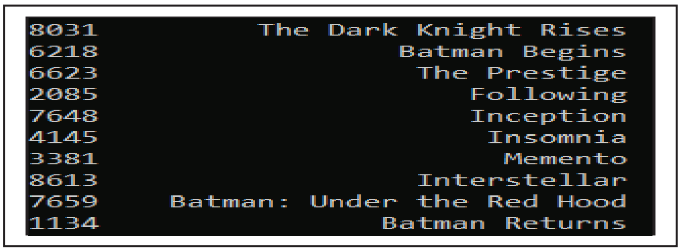


```
7931                          The Dark Knight Rises
132                                  Batman Forever
1113                                 Batman Returns
8227     Batman: The Dark Knight Returns, Part 2
7565              Batman: Under the Red Hood
524                                          Batman
7901                           Batman: Year One
2579              Batman: Mask of the Phantasm
2696                                            JFK
8165     Batman: The Dark Knight Returns, Part 1
```

Fig. 4. Top Recommendations for movies based on content-based filtering approach – using metadata as overviews and taglines.

We see that for “the Dark Knight” our system is able to learn it as a Batman movie and recommend users with movies which are similar of this type.

However, this does not take into consideration the important features like the cast, crew, director or genre of the movie. Thus, we build the next one using the metadata – cast, crew, etc. Results displays recommendations as shown in Fig. 5 below.

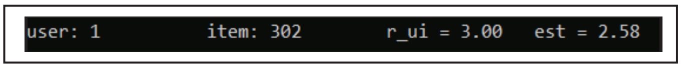


```
8031         The Dark Knight Rises
6218                 Batman Begins
6623                  The Prestige
2085                     Following
7648                     Inception
4145                      Insomnia
3381                       Memento
8613                  Interstellar
7659    Batman: Under the Red Hood
1134                Batman Returns
```

Fig. 5. Top Recommednations for movies based on content-based filtering approach – using metadata as cast, crew, keywords and genre.

B. Predicting Movie Ratings using Collaborative-based filtering approach: -

For movie ID 302, we see an estimated prediction of 2.58. Thus, if we pass the user ID and movie ID, we can retrieve the rating that is user is likely to provide to a particular movie based on his previous ratings for movies of similar genre.



```
user: 1          item: 302          r_ui = 3.00    est = 2.58
```

Fig. 6. Top Recommendations for movies based on collaborative-based filtering approach.

C. Evaluating The Recommendations Provided

We can evaluate the accuracy of the Simple and Content-based filtering algorithms using different metrics. Some evaluation metrics include the Mean Absolute Error (MAE), Percentage Error, Mean Squared Error (MSE) and Root Mean Square Error (RMSE).

a) Mean Absolute Error (MAE):

The MAE is calculated as the mean of the absolute value of the error. The error is the difference between the true value and the predicted value for each of the readings. Mathematically it can be represented as follows using Equation (2):

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} X\,|y - \bar{y}|$$

b) Percentage Error:

The percentage error is calculated as the ratio of the MAE to the true value. Mathematically it can be represented as follows using Equation (3):

$$\text{Percentage Error} = \frac{MAE}{True\ value}\ x\ 100$$

c) Mean Squared Error (MSE):

The MSE is calculated as the mean of the squares of the errors for each of the reading. Mathematically it can be represented as follows using Equation (4):

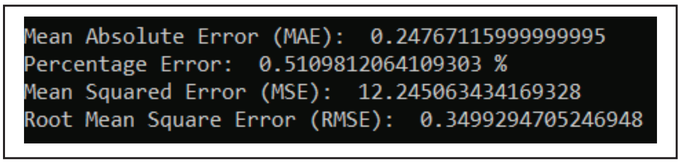
$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y - \bar{y}){}^\wedge 2$$

d) Root Mean Square Error (RMSE):

The RMSE is calculated as the square root of the MSE. Mathematically it can be represented as follows using Equation (5):

$$\text{RMSE} = \sqrt{MSE}$$

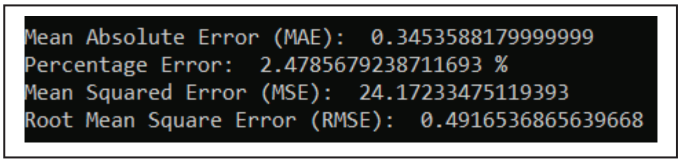
The MAE, percentage error, MSE and RMSE for Simple recommender approach is as shown in Fig. 7 below.



```
Mean Absolute Error (MAE):  0.24767115999999995
Percentage Error:  0.5109812064109303 %
Mean Squared Error (MSE):  12.245063434169328
Root Mean Square Error (RMSE):  0.3499294705246948
```

Fig. 7. Evaluation results for simple recommender approach.

The MAE, percentage error, MSE and RMSE for content-based filtering approach is as shown in Fig. 8 below.



```
Mean Absolute Error (MAE):  0.3453588179999999
Percentage Error:  2.4785679238711693 %
Mean Squared Error (MSE):  24.17233475119393
Root Mean Square Error (RMSE):  0.4916536865639668
```

Fig. 8. Evaluation results for content-based filtering approach.

XI. Conclusion

This paper presents recommendation system for movies using different approaches in Python programming paradigm. It takes the data set as input consisting different movies release on or before July 2017 and provides recommendations based on different approaches.

All the recommender systems implemented have their own advantages and disadvantages in performing their task. Most of the shortcomings in one approach is being taken care by another approach. A good recommendation system should provide positive and useful recommendations and also provide recommendations different from what an existing algorithm already provides.

XII. Future Enhancements

New features can be added to this existing version of project as and when required.

This application can be deployed over the cloud so that it can be accessible to users on a large scale. Elastic Beanstalk environment from AWS can be used to perform these tasks.

Deep Learning can be used to implement recommender systems like – we can make use of neural networks to predict the ratings of users, we can predict the next action that the user is most likely to perform based on his past browsing history, etc.

Future recommendation systems should be able to provide dynamic recommendations. Also, user profiles should be capable of being updated in real-time. However, implementing these calls for the need of huge amount of network bandwidth, computational power, etc. Therefore, new algorithms and techniques should be able to reduce the consumption of network bandwidth, memory, computational power, etc.

Acknowledgment

I take this opportunity to express my sincere gratitude towards the college, Thakur College of Science and Commerce for providing me with an excellent infrastructure to carry out my project activities.

I would like to thank the Principal, Dr. (Mrs.) C. T. Chakraborty for her inspiring and encouraging words.

I am thankful to the IT Department and the Head of the Department, Dr. (Mr.) Santosh Kumar Singh for their kind co-operation towards the completion of this project.

A special vote of thanks and heartfelt gratitude to my project guides, Prof. (Mr.) Mahendra Sharma and Dr. (Mr.) Santosh Kumar Singh for their sincere and encouraging contribution throughout the project span.

References

[1] Sajal Haldar, A. M. Jehad Sarkar, Young-Koo Lee, “Movie Recommendation System Based on Movie Swarm”, 2012 Second International Conference on Cloud and Green Computing, 2012 IEEE, DOI 10.1109/CGC.2012.121, pp.804-809.

[2] Sidnooma Christian Kabore, “Design and implementation of a recommender system as a module for Liferay portal”, Barcelona School of Computing (FIB), University Polytechnic of Catalunya (UPC), September 2012.

[3] P. N. Vijaya Kumar, Dr. V. Raghunatha Reddy, “A Survey on Recommender Systems (RSS) and Its Applications”, International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE), Vol. 2, Issue 8, August 2014, pp.5254-5260.

[4] Bhumika Bhatt, Prof. Premal J Patel, Prof. Hetal Gaudani, “A Review Paper on Machine Learning Based Recommendation System”, International Journal of Engineering Development and Research (IJEDR), Volume 2, Issue 4, 2014, ISSN – 2321-9939, pp. 3955-3961.

[5] Rupali Hande, Ajinkya Gutti, Kevin Shah, Jeet Gandhi, Vrushal Kamtikar, “MOVIEMENDER – A MOVIE RECOMMENDER SYSTEM”, International Journal of Engineering Sciences and Research Tecnology (IJESRT), ISSN – 2277-9655, November 2016, pp.469-473.

[6] S. Halder, A. M. J. Sarkar and Y. Lee, “Movie Recommendation System Based on Movie Swarm”, 2012 Second International Conference on Cloud and Green Computing, Xiangtan, 2012, doi: 10.1109/CGC.2012.121, pp.804-809.

[7] N. Vaidya and A. R Khachane, “Recommender systems – the need of the commerce ERA”, 2017 International Conference on Computing Methodologies and Communication (ICCMC), Erode, doi: 10.1109/ICCMC.2017.8282616, pp.100-104.

[8] https://tryolabs.com/blog/introduction-to-recommender-systems/

[9] https://www.analyticsvidhya.com/blog/2015/10/recommendation-engines/

[10] https://www.expertsystem.com/machine-learning-definition/

[11] https://www.tutorialspoint.com/django/django_overview.htm