Scientific Journal of Impact Factor (SJIF): 4.72

e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 4, Issue 3, March -2017

A Facebook Profile Based TV Shows and Movies Recommendation System

Prof S.R Dhore[1], Abhishek Shukla[2], Anil Kumar Pal[3], Manish Kumar[4]

[1,2,3,4] *Department of Computer Engineering, Army Institute of Technology, Pune, India*

Abstract — *Implemented and evaluated different algorithms in the context of developing a recommendation system based on data gathered from Facebook user profiles. In particular, we are looking at a Collaborative Filtering algorithms, a Content Filtering approach, and Naive Bayes, and comparing their performance in terms of standard measures. The algorithms draw from principles and techniques in Machine Learning, Information Retrieval, as well as Graph Theory. The Facebook graph API was used to scrape friend's Facebook profile data. This results in a dataset of Facebook user profiles in XML format, listing different attributes for a particular user. The 'liked' TV show and movies sections act as the labels for our training and test data , and the rest of the sections are used as the supporting attributes.*

Keywords- Recommendation System, Collaborative Filtering, Content Filtering, Naive Bayes, Information Retrieval, Graph Theory.

I. INTRODUCTION

The traditional TV industry is facing threats and challenges due to the development of the mobile internet. This has happened due to the evolution of Big Data which is changing the traditional industry. For traditional TV shows, audience rating is the metrics whether the show is good or not. Therefore, how to improve the audience rating is an urgent issue for traditional TV shows and movies. This paper proposes a TV shows and movies recommendation system. This system is based on the machine learning algorithms which can automatically recommend TV shows and movies to the audience in accordance to their interest.

Recommendation systems are the one which empower users to use their enormous amount of data and make some informed choices in the future. This field of recommender system has gone through a lot of innovation and research. In the same spirit, this project focuses on building a recommendation system based on the data collected from Facebook profiles of several users.

1.1 Goals and Objectives

The main aim of this project is to predict on what genre of TV shows or movies a user is likely to be interested which will be based on their raw Facebook data and then recommending a set of related items to the user.

The objectives of the project are as follows:

1. Gathering of data from various Facebook profiles using Graph API v2.8.
2. Pre-processing the raw data using different filtering techniques.
3. Data Analysis using different Machine learning Algorithms. Building a recommendation system for TV shows based on data collected from Facebook profiles of several users.
4. Performance measurement and comparison of different algorithms.

II. DATA SET FOR THE SYSTEM

Data Set for the proposed system was captured using the Facebook Graph API. The Graph API is the primary way to get data out of, and put data into Facebook's platform. It's a low-level HTTP-based API that you can use to programmatically query data, post new stories, manage ads, upload photos, and perform a variety of other tasks. The Graph API is HTTP-based, so it works with any language that has an HTTP library, such as cURL and urllib.

As the newer Graph API v2.8 has a limited user profile information access policy. So, a user is only allowed to access his/her friend's user profile data. By using a python script we obtained data of our friends' profile. Most of these profiles had very less or no information. So, we discarded the profiles of people who listed less than two "likes". In the end, we had almost 900 user profiles to work with. Out of this we randomly selected 20% to be the test users.

III. PREPROCESSING OF DATA

As an initial step, various filtering techniques were applied on the acquired data i.e. the Facebook user profiles as well as the metadata on TV shows and movies. The preprocessing step is important to be able to treat the entire data uniformly. Following are the filtering techniques that have been applied on the data:

3.1. Parsing and Tokenization
Entire text data obtained was split into tokens with white-spaces as the delimiting set of characters. After this filtering technique we have a stream of tokens available for the further analysis.

3.2. Stop-Word Elimination
Set of the most commonly used English words was used to remove the useless words which further optimized the data set.

3.3. Stemming
A standard Portman stemming algorithm was applied. Stemming refers to a heuristic process that chops off the ends of words in the hope of achieving this goal correctly most of the time, and often includes the removal of derivational affixes.

3.4. Normalizer
The entire user profile text data was converted to lowercase as a part of preprocessing step, and encoded as UTF-8.

IV. EVALUATION METRICS USED
In case of Facebook profile data, the rating strategy for different entities is of unary type- either user likes an item (that makes a positive association) or not. The absence of like for an item can be considered as a dislike or as ignorance of the item by the user. This strategy is quite different from what other platforms like NetFlix, IMDb uses i.e. rating of items in a discrete range for example from one to five stars.

Traditionally, the evaluation of the recommendation system was done on the basis of RMSE i.e. Root Mean Square Error, method. This method of evaluation works well with the data containing items being rated in a discrete range.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(p_i - a_i)^2}$$

For this project, we chose the Precision-Recall-F1 score framework because the unary rating strategy makes the task of recommending TV shows and movies more similar to a classification problem. Here we have defined the evaluation metric i.e. the precision and recall as follows, where r_i is the TV show recommended to the user while the L denotes the set of TV shows liked by the user.

$$Precision = \frac{\sum_i 1\{r_i : r_i \epsilon L\}}{\sum_i 1\{r_i\}}$$

$$Recall = \frac{\sum_i 1\{r_i : r_i \epsilon L\}}{|L|}$$

The F1 score would be calculated based on the computed results of the above two equations. So, the F1 score would be defined as:

$$F1 = 2.\frac{precision.recall}{precision + recall}$$

The precision obtained can be seen as a measure of exactness or quality, whereas recall as the measure of completeness or quantity. In simple terms, we can say that high precision means that the algorithm returned substantially more relevant results than irrelevant ones, while high recall means that the algorithm returned most of the relevant results.

V. COLLABORATIVE FILTERING
Collaborative filtering, also referred to as social filtering, filters information by using the recommendations of other people. It is based on the idea that people who agreed in their evaluation of certain items in the past are likely to agree again in the future. A person who wants to see a movie for example, might ask for recommendations from friends.

The main theme of this algorithm is to build similarities between different users based on the TV shows and movies listed in their Facebook profiles, and between items (TV shows and movies) based on the user interest. This helped us in

recommending TV shows and movies to a user based on the viewing preferences of other similar users whom he/she might or might not be connected to. We have implemented two different variations of collaborative filtering one is User-Based Collaborative Filtering while the other one is Item-Based Collaborative Filtering.

5.1. User-Based Collaborative Filtering

In the implementation of this technique, we used the classic *kNN* algorithm to obtain a neighborhood of users similar to the user who will be recommended the TV shows of interest. In our case, we have used cosine-based similarity.

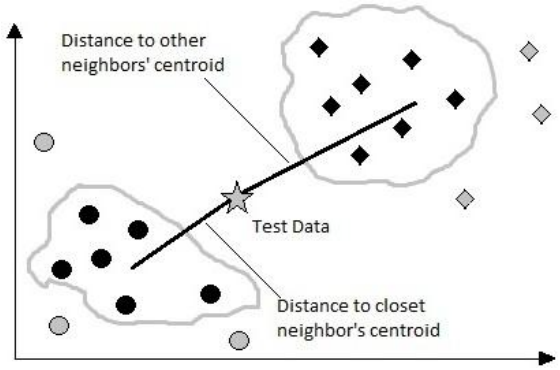

Figure 1: Example of kNN Algorithm

In the cosine-based technique, two items are thought of as two vectors in *m* dimensional user-space. The similarity between them is measured by computing cosine angle between two vectors. Similarity between items *i* and *j*, denoted by *sim(i.j)* is given by

$$sim(i,j) = cos(\vec{i},\vec{j}) = \frac{\vec{i}.\vec{j}}{\|\vec{i}\|\|\vec{j}\|}$$

In our case, the two vectors *i* and *j* can be considered as the user's likes and dislikes or ignorance in unary- based system. A higher value of *sim(i,j)* indicates that the two users are closer in their tastes of TV shows or movies.

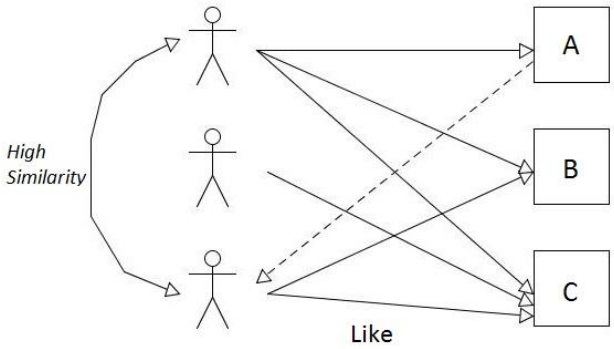

Figure 2: User-Based Collaborative Filtering

5.2. Item-Based Collaborative Filtering

In the technique, computing the similarity between items is the fundamental step of our recommendation system, since we want to recommend similar TV shows and movies to the users based on what they have already watched before. The basic idea of similarity computation between two shows or movies *i* and *j* is to firstly isolate the users who have liked both of these items and then to apply a similarity computation technique to determine the similarity *sim(i,j).*

This algorithm is quite similar to the User-Based Collaborative Filtering technique, with the users and items i.e. TV shows or movies switching their roles. Rather than using similarities between users' likes to predict preferences, item-based collaborative filtering uses similarities between liking patterns of different TV shows or movies.

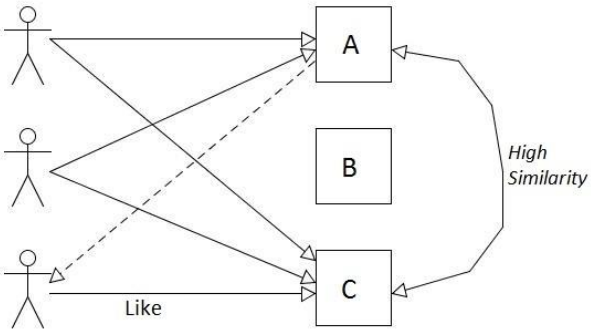

Figure 3: Item-Based Collaborative Filtering

If two TV shows or movies have the same set of user's likes, then they are similar, users are expected to have similar preferences for similar items. Our implementation used the same cosine-based similarity measure to obtain the similarity between two items.

We have performed 10-fold cross-validation on each of the above mentioned approaches in order to evaluate performance.

VI. CONTENT-BASED FILTERING

Content-based filtering technique, recommends items based on a comparison between the content of the items and a user's profile. The content of each item is represented as a set of descriptors or terms. The user's profile is represented with the help of same terms and built up by analyzing the content of items which the user has already seen.

Several issues need attention while implementing a content-based filtering system. First, descriptors or terms can either be assigned automatically or manually. When terms are assigned automatically a method has to be chosen that can extract these terms from items. Second, the terms have to be represented in such a way that both the user profile and the items can be compared in a meaningful way. Third, a learning algorithm has to be chosen that is able to learn the user's profile based on seen items and can make recommendations based on this user profile.

Relevance feedback, genetic algorithms, neural networks, and the Bayesian classifier are among the learning techniques for learning a user profile.

In our case, after preprocessing the acquired Facebook user's profile data through various filtering techniques mentioned in the Section III., content-based filtering techniques were being applied. Following image depicts the basic work-flow of the implementation:

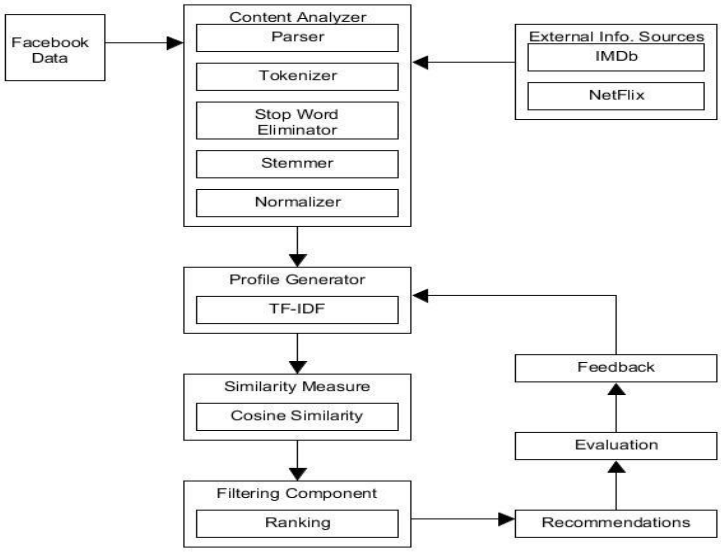

Figure 4: Content-Based Filtering Strategy

Next after preprocessing stage, tf-idf scores were being computed for the different features and then the cosine-based similarity technique as mentioned in Equation in Section IV was used in the ranking function to compute the items (i.e.

TV shows or movies) most similar to the test user. The features for particular item that can be accumulated may be like Genre, Director, Actors or description via certain related keywords. Now, the recommendations were done on basis of ratings of items in the test subject's neighborhood.

Genre/ Movies	Sci-fi	Action	Comedy	Children	Romance	User(Bob)
Transformer	X	X				+
Over the Hedge				X		+
Titanic					X	+
The Princess Bride			X		X	+
Taken		X				?

Figure 5: Example of our case

In order to generate results with more accuracy, we have used only those users who had liked at least two or more shows for which we had some metadata, collected from an external source like Screaming Velocity or IMDb. A 70-30 %split of this reduced set of users was used to carry out cross-validation.

VII. CONCLUSION

After trying out these different approaches it became clear that predicting movie and TV show interests based on information in Facebook profiles is a very complex task. The highest precision we got was still around 60%. We faced various challenges while implementing this project. The sparsity of the data, notably the fact that most profiles did not include movie and TV show interests, made it more difficult for learning algorithms to predict that a particular profile would like the show that we were trying to predict. A similar problem occurred with our Collaborative and Content-Based algorithm, in this case the lack of similarities between profiles caused problems. For both of these types of algorithms, we saw that testing on profiles that have at least two movies or TV shows interest increased their precision.

A recommendation system takes a lot of effort to build and fine-tune. It took an immense amount of labor to collect, process, filter, and organize the data before we could run any algorithms. The algorithms that used external data and attributes from a user's Facebook profile performed consistently better than those that didn't avail any metadata.

REFERENCES

[1] Mengyi Zhang, Minyong shi, Zhiguo Hong, Songtao Shang and Menghan Yan, "A TV Program Recommendation System Based on Big Data" .ICIS 2016, June 26-29, 2016, Okayama, Japan.

[2] Zhaocai Ma, Yi Yang, Fei Wang, Caihong Li, Lian Li, "The SOM based Improved K-means Clustering Collaborative Filtering Algorithm in TV Recommendation System", 2014 Second International Conference on Advanced Cloud and Big Data.

[3] Alejandro Ayala-Hurtado, Yeskendir Kassenov, Nick Yannacone, "Creating TV Movie Recommendations From Facebook Profile Information"

[4] Jin An Xu, Kenji Araki, "A SVM-based Personal Recommendation System for TV Programs.",1-4244-0028-7/06 ©2006 IEEE

[5] Johan Lindell, Anders Haponen, "Predicting Movie and TV Preferences from Facebook Profiles".

[6] Mathangi Venkatesan,Andy Mai , "Recommendation of TV shows and Movies based on Facebook data".

[7] *http://scikit-learn.org/stable/documentation.html*

