Movie Recommendation System Using Content-Based Filtering

Nikitha Chennamaneni, Laxmisravya Nimmagadda
The University of Texas at Arlington (UTA), Texas, United States
laxmisravya.nimmagadda@mavs.uta.edu,nikitha.chennamaneni@mavs.uta.edu

Abstract

In today's world of internet, Recommendation systems play a major role in our day to day life by providing suggestions to users for certain resources like movies, books, education, online shopping, etc. It has the ability to determine whether a user would prefer an item or not, based on the user's profile. A recommendation system is one of the widespread applications of machine learning that deals with the tendency of a certain user towards an item based on his/her likeliness towards it previously. Many factors can be considered while developing a movie recommender system like genre, ratings, tags, feedback, director of the movie, and many more. In this project, recommendation systems for the movies are developed. Movies recommendation systems usually predict what movies a user will like based on the attributes related to the previous search history or liked movies. The recommender movie system can recommend a movie by a combination of two or more attributes. The proposed system helps the user in picking movies, where the user's search input genre, ratings, and tags would help to predict the movies for that particular user. The approach adopted to do so is content-based filtering using genre correlation.

Introduction

Nowadays, there has been immense growth in the amount of digital information and the number of users as well. Also, the quantity of data transactions on the internet has drastically increased. This triggered a potential challenge of information overload which obstruct timely access to the available data on the internet. With the number of users is increasing, the amount of data is increasing too dramatically. The problem arises when a user has to search for long periods to obtain their desired movie or information. The recommendation system helps in solving this problem. The recommendation system helps the user to find their desirable item by predicting their previous performance and suggesting relevant information to the user. They are essentially a central part of websites like movies, music, e-commerce applications, and many more. Recommendation systems help in addressing information overload problems by retrieving desired information of an individual so that next time it helps that particular user in searching for desired movies in lesser time.

Content-based filtering using correlation is compatible with other preference-based services like education services, where information on preferred subjects or instructors would be seamlessly applicable. K-means clustering is rather easy to apply to even large data sets, particularly when using heuristics such as Lloyd's algorithm. It has been successfully used in market segmentation, computer vision, and astronomy among many other domains.

Dataset Description

Data Description:

This dataset (ml-25m) describes 5-star rating and free-text tagging activity from Movie-Lens, a movie recommendation service. It contains 25000095 ratings and 1093360 tag applications across 62423 movies. All selected users had rated at least 20 movies. No demographic information is included. Each user is represented by an id, and no other information is provided. The data is contained in the files movies.csv, ratings.csv, and tags.csv.

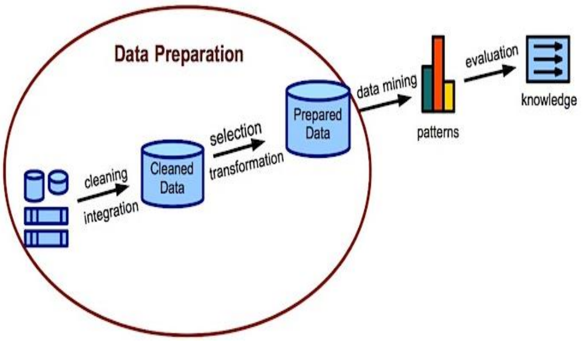
Data Pre-Processing:

Pre-processing refers to the transformations applied to our data before feeding it to the algorithm. Data Pre-processing is a technique that is used to convert the raw data into a clean data set of an understandable format. Data in the real world is incomplete, noisy, and inconsistent. Data that is not scaled and standardized might have an unacceptable prediction. The more disciplined you are in your handling of data, the more consistent and better results you are likely to achieve. Machine learning data is as good as it is built from. For the dataset we have taken, we don't need to do the rescaling as the data in the attributes will be either an integer or a string such as a movie rating which is an integer and movie name attribute which is a string.

Project Description

1. Description

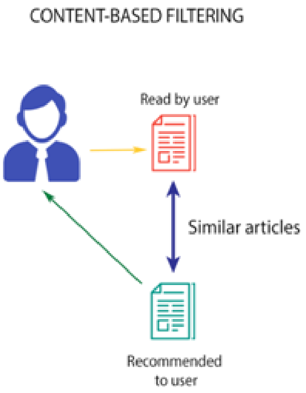
In this project, we recommend movies to the users based on the history of their personalized searches and reviews. Both people and online streaming services need a user based personalized movie recommendation system which helps in analyzing the interests of each individual and recommends the best movies suited to them. The approach used for building the movie recommendation system is content-based filtering. As we know that content-based filtering analyses the user's past behavior and recommends items similar to it based on the parameters considered. In this Content-based movie recommender system gives their recommendations based on attributes like Ratings, Genre, and Tags. If a user has rated highly for a certain movie, other movies containing similar genres and then followed by tags are recommended by the user. Another Enhancement approach is Collaborative Filtering using K-Means Clustering making an automatic prediction (filtering) about the interests of a user by collecting interests from many related users.



The above figure shows the complete process of a recommender system

2. Recommendation System Using Content-Based Filtering

A recommender system is an information filtering technique with two major paradigms. There are collaborative and content-based filtering methods. Content-based recommender systems try to recommend items like those a given user has liked in the past.



This system uses content-based filtering for recommending movies to users based on the similarity of genres, ratings, and tags. The dataset used for this purpose is movies, ratings, and tags.

Implementation of the proposed algorithm

For a particular movie, the recommendations are given based on rating first. Considering the average rating of the movie, movies with a similar average rating are recommended to the user. Obtained movies list is further sorted based on genres and movies with a similar genre are recommended to the users. Then followed by tags are recommended by the user.

Below algorithm is implemented for extracting top 30 movies based on the ratings

Content-Based Movie Recommendation System Based on Ratings

Step1. Reading the movies, rating datasets that are taken after performing data pre-processing

Step2. Sorting the required columns from all the datasets

Step3. Merging the rating dataset with movie names and finding the average rating of each movie and the number of ratings for each movie

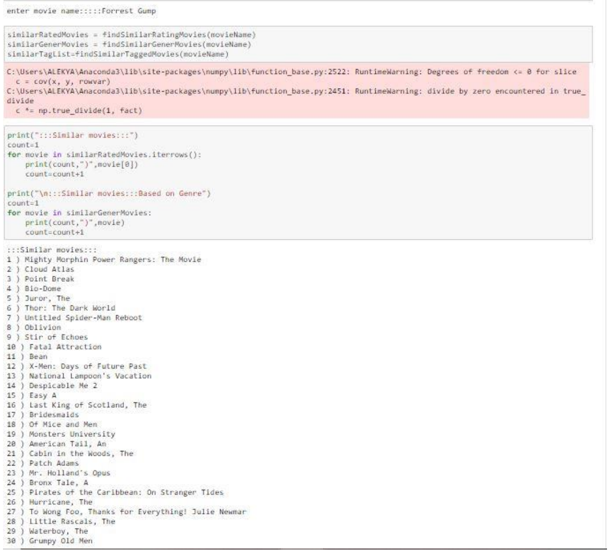
Step4. Creating a matrix with userid and title as axis and rating as data

Step5. Correlating the user accessed movie with other movies

Step6. Converting the results into a dataframe for further analysis

Step7. Merging the movie correlation with the count of each movie

Step8. Return the top 30 similar rated movies. These are the recommended movies for the current user.



```
enter movie name::::Forrest Gump
```

```
similarRatedMovies = findSimilarRatingMovies(movieName)
similarGenerMovies = findSimilarGenerMovies(movieName)
similarTagList=findSimilarTaggedMovies(movieName)
```

```
C:\Users\ALEKYA\Anaconda3\lib\site-packages\numpy\lib\function_base.py:2522: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar)
C:\Users\ALEKYA\Anaconda3\lib\site-packages\numpy\lib\function_base.py:2451: RuntimeWarning: divide by zero encountered in true_
divide
  c *= np.true_divide(1, fact)
```

```
print(":::Similar movies:::")
count=1
for movie in similarRatedMovies.iterrows():
    print(count,")",movie[0])
    count=count+1

print("\n:::Similar movies:::Based on Genre")
count=1
for movie in similarGenerMovies:
    print(count,")",movie)
    count=count+1
```

```
:::Similar movies:::
1 ) Mighty Morphin Power Rangers: The Movie
2 ) Cloud Atlas
3 ) Point Break
4 ) Bio-Dome
5 ) Juror, The
6 ) Thor: The Dark World
7 ) Untitled Spider-Man Reboot
8 ) Oblivion
9 ) Stir of Echoes
10 ) Fatal Attraction
11 ) Bean
12 ) X-Men: Days of Future Past
13 ) National Lampoon's Vacation
14 ) Despicable Me 2
15 ) Easy A
16 ) Last King of Scotland, The
17 ) Bridesmaids
18 ) Of Mice and Men
19 ) Monsters University
20 ) American Tail, An
21 ) Cabin in the Woods, The
22 ) Patch Adams
23 ) Mr. Holland's Opus
24 ) Bronx Tale, A
25 ) Pirates of the Caribbean: On Stranger Tides
26 ) Hurricane, The
27 ) To Wong Foo, Thanks for Everything! Julie Newmar
28 ) Little Rascals, The
29 ) Waterboy, The
30 ) Grumpy Old Men
```

Below algorithm is implemented for extracting top 10 movies based on the genre by considering the rating algorithm outcome

Content-Based Movie Recommendation System based on Genre

Step1. Reading the movies, rating datasets that are taken after performing data pre-processing

Step2. Sorting the required columns from all the datasets and obtaining the genre of the user accessed movie

Step3. Iterating through all similar rated movie list and obtaining the genre for each movie

Step4. Comparing user accessed movie genre list with similar rated movie genre

Step5. Sorting based on genre match count

Step6. Return the top 10 movies. These are the recommended movies for the current user.

Below algorithm is implemented for extracting top movies based on the tags by considering the genre algorithm outcome

Content-Based Movie Recommendation System based on Tags

Step1. Reading the movies, rating datasets that are taken after performing data pre-processing

Step2. Previously recommendation was performed based on ratings and genres and the result is used for recommending movies based on tags.

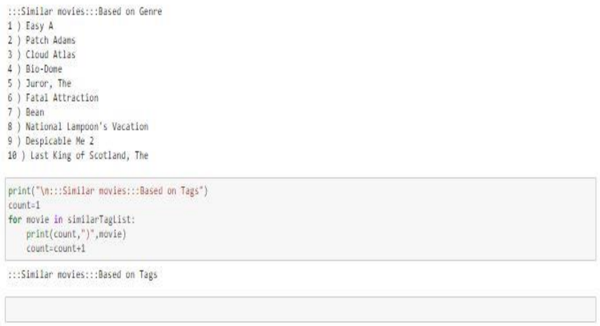
Step3. First, we retrieve similar genre movies list and get tags list of searches.

Step4. Iterating through all similar genre movies and comparing them with tags of search.

Step5. The tag match count is calculated and sorted accordingly.

Step6. Return the movies for which tag match.

These are the recommended movies for the current user.



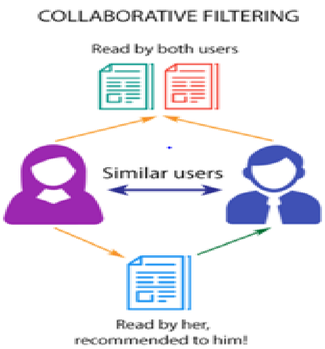
```
:::Similar movies:::Based on Genre
1 ) Easy A
2 ) Patch Adams
3 ) Cloud Atlas
4 ) Bio-Dome
5 ) Juror, The
6 ) Fatal Attraction
7 ) Bean
8 ) National Lampoon's Vacation
9 ) Despicable Me 2
10 ) Last King of Scotland, The
```

```
print("\n:::Similar movies:::Based on Tags")
count=1
for movie in similarTagList:
    print(count,")",movie)
    count=count+1
```

```
:::Similar movies:::Based on Tags
```

3. Extension- Collaboration-Based Movie Recommendation System Using K-Means Clustering

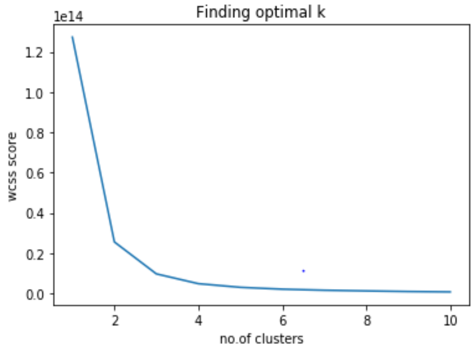
Collaborative Filtering is an approach of making an automatic prediction (filtering) about the interests of a user by collecting interests from many related users. The purpose of clustering is to partition objects into groups known as clusters in such a way those two objects within the same cluster have a minimum distance between them to identify similar objects then the clustering process is performed offline to build the model. When a target user arrived, the online module allocates a cluster with a substantial similarity weight to the user, and the prediction rating of a specified item is computed based on the same cluster members instead of searching whole user space. K represents the number of clusters. The system takes in the user's personal information and predicts their movie preferences. Afterward, it clusters the movies and generates questions to refine the recommendation. Finally, it suggests movies for users.



What is K-Means Clustering?

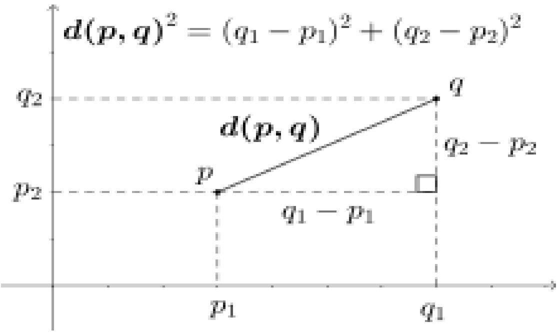
K-Means algorithm is an iterative algorithm that partition the dataset into K pre-defined distinct non-overlapping subgroups (clusters) where the data belongs to only one particular group. In this project K – Value is evaluated by using the elbow method to find the optimal value of k to be used in k-Means. The graph is plotted against k value and Within-Cluster-Sum-of-Squares (WCSS) at each value of k and Euclidean Distance is used to calculate the distance from the centroid.

Graph Plot



What is Euclidean Distance?

Similarity measures between two users can be found by Euclidean, Manhattan, cosine distance. In our project, we used Euclidean distance and the formula for it is as ‘below’.



The Standard K-means Clustering Algorithm:

Step 1: Place K points into the space specialized by the users that are being clustered. These points represent an initial set of centroids.

Step 2: Assign each user to the group (cluster) that has the closest centroid.

Step 3: When all users have been assigned, recalculate positions of K centroids for each cluster.

Step 4: Repeat steps two and three until the centroids no longer move. This produces a separation of the users into a group (cluster) from which the metric to be minimized can be calculated.

Proposed Algorithm for Collaborative-Based Movie Recommendation System Using K-Means Clustering:

Step 1: Reading the movies, rating datasets that are taken after performing data pre-processing then re-index and merge the datasets

Step 2: Elbow method is used to find the optimal value of k to be used in k-means.

Step 3: Calculate the squared sum for each value of k within the cluster from their cluster centroid, called as Within-Cluster-Sum-of-Squares(WCSS)

Step 4: Plotting a graph against k value and wcss at each value of k.

Step 5: Define the k-means object with the number of clusters and fitting the data to it.

Step 6: Tag the rating dataset with the cluster information

Step 7: Calculate the number of times each movie is differently labeled and finally classifies the movies cluster as the label having the highest count.

Step 8: Rename the data frame columns and merge movieId with the obtained data frame

Main references used for your project

In [1] we get to know about the proposed algorithm where dot product and similarity measures are used to implement correlation functions.

In [2] we learned about how to process k-means and collaborative filtering.

The difference in ACCURACY/PERFORMANCE between your project and the main projects of your references

When it comes to performance analysis, depending on the Correlation algorithm and K-Means the Performance of Correlation is more efficient when compared to K-Means clustering because it correlates between movies whereas K-means takes a random cluster of samples each time and each time the recommendation set of movies differs. By the nature of the system, it is not a straightforward task to evaluate the performance since there is no right or wrong recommendation; it is just a matter of opinion.

List of your contributions in the project (your work)

1. Pre-processing the required data
2. Implementation of Content-based learning using Correlation
3. Implementation of Collaborative-based learning from scratch using K-means algorithm
4. Theoretical comparison of the above algorithms

Analysis

1. What did I do well?

Generally, the content-based movie recommender systems give their recommendations based on the attributes like movie name, cast, Director, Genre. But In our project, we can get the recommendations based on the storyline because we have added an overview of the movie attribute which facilitates finding the similarity between the inputs which is given by the user. Whereas in K –Means clustering allowed us to approach a domain without really knowing a whole lot about it and draw conclusions. It let us do that by learning the underlying patterns in the data for us, only asking that we gave it the data in the correct format.

2. What could I have done better?

A hybrid approach of two main algorithms (Content-Based Filtering and Collaborative Filtering) together with Correlation Coefficient in providing recommendations would yield better recommendations for the users and mathematical comparison among the two approaches is yet to be evaluated but since we did not pre-process the data according to the training and testing approach the comparison or they can be differentiated theoretically.

3. What is left for future work?

A possible direction of future study will extend to the applicability of customer feedback on the recommendation performance of the proposed model. We would like to implement a Web-based user interface that has a user database and has the learning model tailored to each user.

Conclusion

In this paper, we proposed content-based filtering using genre correlation and we implemented K-means clustering as well. The recommendation system implemented in this paper aims at providing movie recommendations based on the genres of the movies, ratings, and tags. There is no right or wrong recommendation because it depends on the user's opinion, so it is not a straightforward task to evaluate the performance. The difference between both filterings using correlation has issues of data sparsity and scalability where clustering reduces the problem of scalability. The advantage of collaborative filtering is that it does not rely on content that can be analyzed and can accurately represent complex items.

The results show that the collaborative filtering recommendation system is more suitable in recommending movies to active users because this system is more successful in producing desirable recommendations compared to the content-based recommendation system.

References

[1]. Content-Based Movie Recommendation System Using Genre Correlationhttps://www.researchgate.net/publication/331966843_Content-Based_Movie_Recommendation_System_Using_Genre_Correlation

[2]. Movies recommendation system using collaborative filtering and k-means https://www.researchgate.net/publication/314250702_Movies_recommendation_system_using_collaborative_filtering_and_k-means/link/58e4a8c00f7e9bbe9c94dbb4/download

[3]. The Use of Collaborative Filtering, Content-based Filtering and Pearson Correlation Coefficient for Multilevel Recommender System http://www.acadpubl.eu/hub/2018-118-21/articles/21b/56.pdf

[4]. A Movie Recommendation System http://www.ijesrt.com/issues%20pdf%20file/Archive-2016/November-2016/63.pdf

[5]. Using Content-based filtering for Recommendation http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.25.5743&rep=rep1&type=pdf

[6]. Content-based Recommender systems: State of Art and Trends http://facweb.cs.depaul.edu/mobasher/classes/ect584/Papers/ContentBasedRS.pdf

[7]. The Use of Collaborative Filtering, Content-based Filtering and Pearson Correlation Coefficient for Multilevel Recommender System http://www.acadpubl.eu/hub/2018-118-21/articles/21b/56.pdf

[8]. Recommender system design using movie genre similarity and preferred genres in SmartPhone

[9]. The 4 Recommendation Engines That Can Predict Your Movie Tastes https://towardsdatascience.com/the-4-recommendation-engines-that-can-predict-your-movie-tastes-109dc4e10c52