Title: Recommender systems on Netflix dataset.

(Content Based Recommender System and Collaborative Filtering Recommender System)

Abstract

Recommendation System belongs to the class of Information Retrieval, Data Mining and Machine Learning. Recommender systems play a major role in today's ecommerce industry. Recommender systems recommend items to users such as books, movies, videos, electronic products and many other products in general. Recommender systems help the users to get personalized recommendations, helps users to take correct decisions in their online transactions, increase sales and redefine the users web browsing experience, retain the customers, enhance their shopping experience. Information overload problem is solved by search engines, but they do not provide personalization of data. Recommendation engines provide personalization. There is different type of recommender systems such as content-based, collaborative filtering, hybrid recommender system, demographic and keyword-based recommender system. Variety of algorithms are used by various researchers in each type of recommendation system. Lot of work has been done on this topic, still it is a very favorite topic among data scientists. It also comes under the domain of data Science.

In this project, we will be exploring two methods of recommender systems which are Content Based Recommender system and Collaborative Filtering Recommendation System, using two separate datasets, one for each. The content-based recommender system makes predictions on the content to be recommended to the user based on the initial content provided. Collaborative filtering recommender system on the other hand makes predictions of user ratings for the existing content.

Introduction

A recommender system is an information filtering system which recommends the products, items or content to the user.

Recommender systems are used to recommend a variety of products such as movies, music, news, books, research articles, restaurants, historic places, grocery stores, shopping malls, Twitter pages and many more. As we can see, recommender systems are being used by Amazon, Flipcart.com, Twitter, Netflix and Youtube.com

In the days before internet, people found difficult as to which book to read, which movie to watch or which place to visit. They decided by listening to the comments given by their friends. It is from this concept that recommendation systems were born. Rather than getting recommendations from unknown people and strangers’ users are more comfortable to get recommendations from their own interests and recommendations from users based on similar interests.

Variety of information is available on the web. Some people who lack experience can make poor decisions if recommendations are not available. Because of recommender systems the visitors who browse the websites end up buying the product or watching the content and they get satisfaction. The more the customers are satisfied with the recommender systems they use it more and more popular, becomes the website.

In a recommendation system we have an item and a user. Every recommender system needs some parameters such as rating or item attributes in order to recommend the product to the user.

Challenge is to provide high quality and accurate recommendations, make recommendations for lots of people in a very less time and also increase the hit rate on the website.

Like many machine learning techniques, a recommender system makes prediction based on users' historical behaviors. Specifically, it's to predict user preference for a set of items based on past experience. To build a recommender system, the most two popular approaches are Content-based and Collaborative Filtering.

Content-based 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 extracted by applying Natural Language Processing.

Collaborative Filtering, on the other hand, 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. In terms of user preference, it usually expressed by two categories. *Explicit Rating*, is a rate given by a user to an item on a sliding scale, like 5 stars for Titanic. This is the most direct feedback from users to show how much they like an item. *Implicit Rating*, suggests users' preference indirectly, such as page views, clicks, purchase records, whether or not listen to a music track, and so on.

In this project, we will take a close look at collaborative filtering (which uses Explicit rating) and content-based approach for recommender systems.

The rest of the paper is organized as follows: Section II is an overview of recommender systems; Section III describes the data, analysis of the data and the details of the methodology for this work; Section IV covers the implementation and the results; and in Section V draws our conclusions.

Section II

Background

Recommender Systems

A recommender system is a subclass of information filtering system that seeks to predict the "rating" or "preference" a user would give to an item. They are heavily used in many commercial applications such as Netflix, Youtube, and Amazon Prime. A recommender system helps users to find relevant items in a fraction of time and without the need to search the entire dataset.

A. **Content Based Recommender Systems**
This approach utilizes the properties and the metadata of a particular item to suggest other items with similar characteristics. For example, a recommender can analyze a movie's genre and director to recommend additional movies with similar properties. Content based recommender system [7] type compares the already purchased or searched items by the user and recommends similar items to the user. When we are asking our friends what movie to see, which restaurant to go, what book to read we are taking recommendations from them. To increase the performance of recommendation process social relationship of user needs to be considered. A pure content-based recommender system only considers user's choices, a system needs to consider the choices of user's friends also [8]. This can increase the prediction accuracy and give satisfaction to the user.

In this project, we are using TF-IDF matrix method and a count vectorizer method to make recommendations, which capture the important genres of each movie by giving a higher weight to the less frequent genres.

B. Collaborative Filtering Recommender Systems

Ecommerce websites like, Amazon use collaborative filtering recommender system. A person for example wants to watch a movie, he will ask his friends. Recommendations from people who have similar interests are always better than others.

A number of users are selected which are similar to the active user ie. the user who wants recommendation. A prediction for the active user is done by calculating the weighted average of all ratings of similar users. Accuracy of a recommendation system depends on how best the other users are selected, which are similar to the user who wants recommendation. In this project, gradient descent is used to reduce the cost function and the root mean squared error is also calculated and the user ratings are predicted, based on which the movie recommendations are made. RMSE for the probe dataset is calculated and used as an evaluation metric. Comparison was made with the lowest Cost/RMSE we could get with our gradient descent algorithm with the results generated from passing the cost function and gradients into a highly optimized cost-reduction function called fmincg.

Section III

Methods

In this section I cover the methodology for this work. I have highlighted the dataset that I have used and how I handled them, the feature space I decided to explore and the recommender systems that were focused upon.

A. Content Based Recommender System Dataset

The Netflix dataset has been used in the Content Based Recommender System.

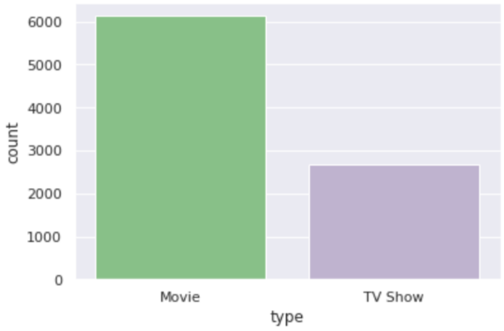
Netflix is one of the most popular media and video streaming platforms. They have over 8000 movies or tv shows available on their platform, as of mid-2021, they have over 200M Subscribers globally. This tabular dataset consists of listings of all the movies and tv shows available on Netflix, along with details of the respective columns: show_id, type, title, director, cast, country, date_added, release_year, rating, duration, listed_in, description.

The ‘type’ consists of two kinds of shows: Movies and TV Shows.

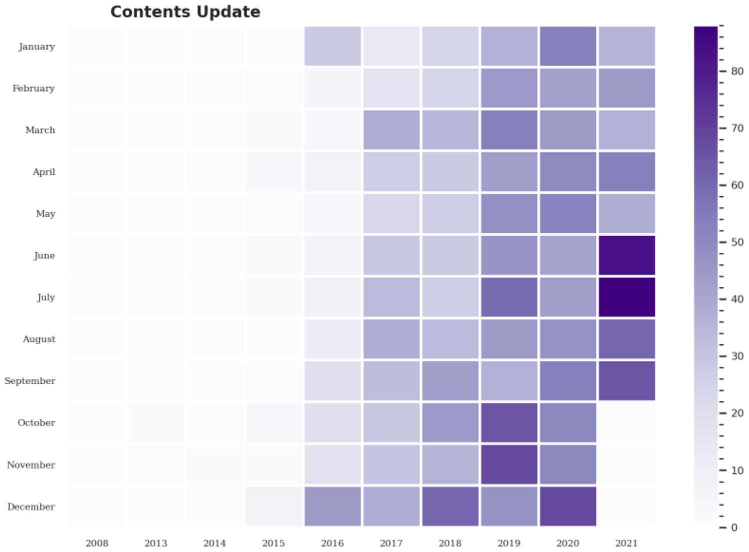
Analysis of Data:

Here are some of the plots that were plotted to understand the data in a better way.

a. The below plot shows the count of the number of movies and the number of TV Shows in the dataset.

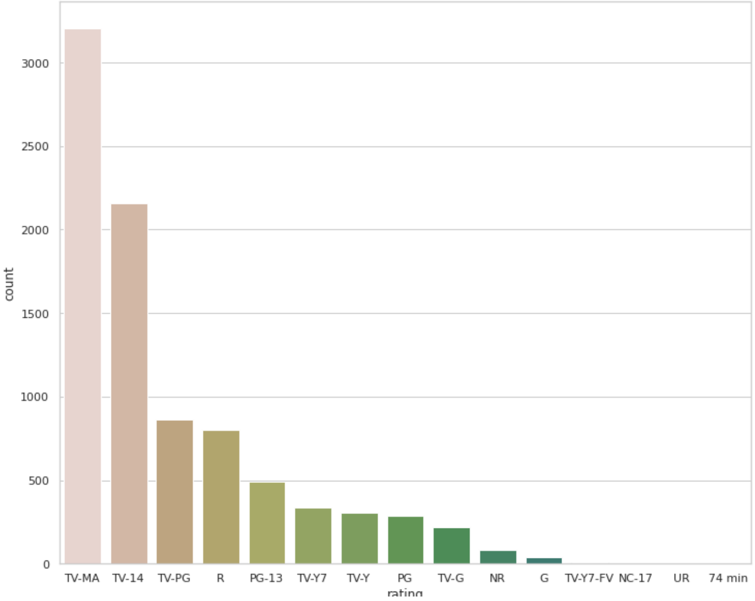


b. The below graph captures the content of Netflix.



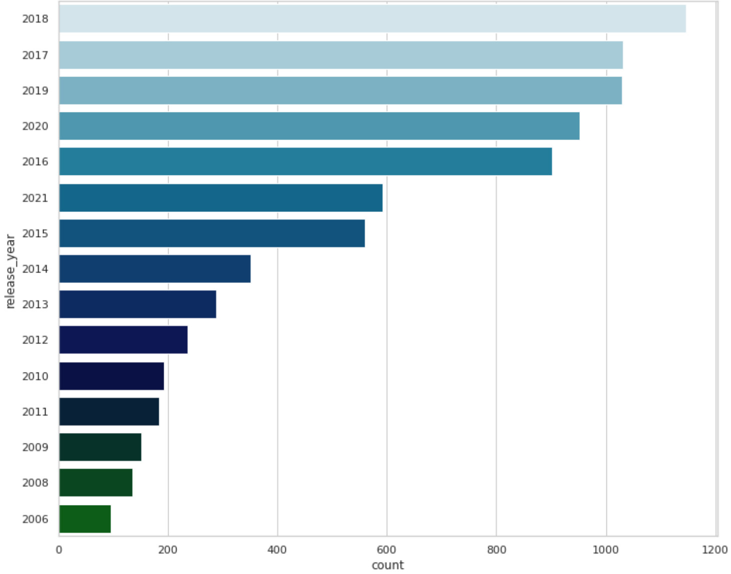
__Observation:__ In this plot the darker the color gets, the higher the number of contents rises. Therefore, we can observe that the number of contents was at its highest in June and July 2021.

c. The below graph shows the count of the user ratings of users belonging to different categories.



<u>*Observation:*</u> By this count plot of ratings we can realize that TV-MA (Mature Audiences) contents have the highest rating numbers. Then it gets lower as in order TV-14 (Material that parents or adult guardians may find unsuitable for children under the #age of 14) and TV-PG (Parental Guidelines).

d. *The below graph shows the count of the shows released in each year.*



<u>*Observation:*</u> This plot shows that most of the contents were produced in 2018 and 2017. Interesting fact is in 2019, 2020 and 2021 the number of produced contents is low. This is mostly because Covid-19 pandemic.

Initially, the Term Frequency (TF) and Inverse Document Frequency was used to determine the relative importance of the movie title based on the user rating. TF-IDF is used mainly because of two reasons: Suppose we search for “the rise of analytics” on Netflix. It is certain that “the” will occur more frequently than “analytics” but the relative importance of analytics is higher than the search query point of view. In such cases, TF-IDF weighting negates the effect of high frequency words in determining the importance of an item (document). The cosine similarity matrix is then calculated to determine the similarity of movies in the recommender system. But these results are not very accurate, hence I have performed Content Based Filtering based on a few features alone using the count vectorized method. Those features include, title, director, cast, listed_in, description. Here the count vectorizer is used instead of tf-idf and hence the results are more accurate.

B. Collaborative Filtering Recommender System Dataset

A different Netflix dataset has been used in the Collaborative Filtering Recommender System.

The data were collected between October, 1998 and December, 2005 and reflect the distribution of all ratings received during this period. The ratings are on a scale from 1 to 5 (integral) stars. The date of each rating and the title and year of release for each movie id are also provided.

a. Training Dataset File Description:

The file "training_set.tar" is a tar of a directory containing 17770 files, one per movie. The first line of each file contains the movie id followed by a colon. Each subsequent line in the file corresponds to a rating from a customer and its date in the following format:

CustomerID,Rating,Date

- CustomerIDs range from 1 to 2649429, with gaps. There are 480189 users.
- Ratings are on a five-star (integral) scale from 1 to 5.
- Dates have the format YYYY-MM-DD.

b. Movies File Description
Movie information in "movie_titles.txt" is in the following format:

MovieID,YearOfRelease,Title

- MovieID do not correspond to actual Netflix movie ids or IMDB movie ids.
- YearOfRelease can range from 1890 to 2005 and may correspond to the release of corresponding DVD, not necessarily its theoretical release.
- Title is the Netflix movie title and may not correspond to titles used on other sites. Titles are in English.

c. The qualifying dataset for the Netflix Prize is contained in the text file "qualifying.txt". It consists of lines indicating a movie id, followed by a colon, and then customer ids and rating dates, one per line for that movie id. The movie and customer ids are contained in the training set. Of course, the ratings are withheld. There are no empty lines in the file.

```
MovieID1:
CustomerID11,Date11
CustomerID12,Date12
```

d. The program predicts the all ratings the customers gave the movies in the qualifying dataset based on the information in the training dataset.

Example of an output:
if the qualifying dataset is:

```
111:
3245,2005-12-19
5666,2005-12-23
6789,2005-03-14
225:
1234,2005-05-26
3456,2005-11-07
```

then the prediction file looks like:

```
111:
3.0
3.4
4.0
225:
```

1.0
2.0

which means that customer 3245 would have rated movie 111 3.0 stars on the 19th of Decemeber, 2005, that customer 5666 would have rated it slightly higher at 3.4 stars on the 23rd of Decemeber, 2005, etc.

e. Implemented random initialization of a θ matrix for movies and a X (feature) matrix for users, normalized our Y (movies x users) matrix, wrote functions to calculate Costs/Gradients, performed gradient descent, and then calculated the final RMSE (root mean squared error) as our evaluation metric.

C. Features

a. Method A: **Content Based Recommender System Dataset**
The dataset was pre-processed in the following ways.
   - Stopwords filtering - Filtering stop words involves pre-processing the review text to remove common words such as "the," "is," etc. This tuning parameter may help remove irrelevant features from the feature set.
   - Replacing NaN with an empty string.
   - Store the Netflix movies and shows separately in two different variables.
   - Drop the null values for date.
   - Store the year and month in two separate variables to make the computation easier.

b. Method B: **Collaborative Filtering Recommender System Dataset**
The dataset was cleaned before proceeding ahead with it.
These were the two actions performed.
   a. New column "Movie_Id" was added to the Data Frame that associates each rating to a movie.
   b. Last record and its corresponding length were recorded.

Section IV

Experiments

Let Method A refer to 'Content Based Recommender System', and Method B refer to 'Collaborative Filtering Recommender System'.

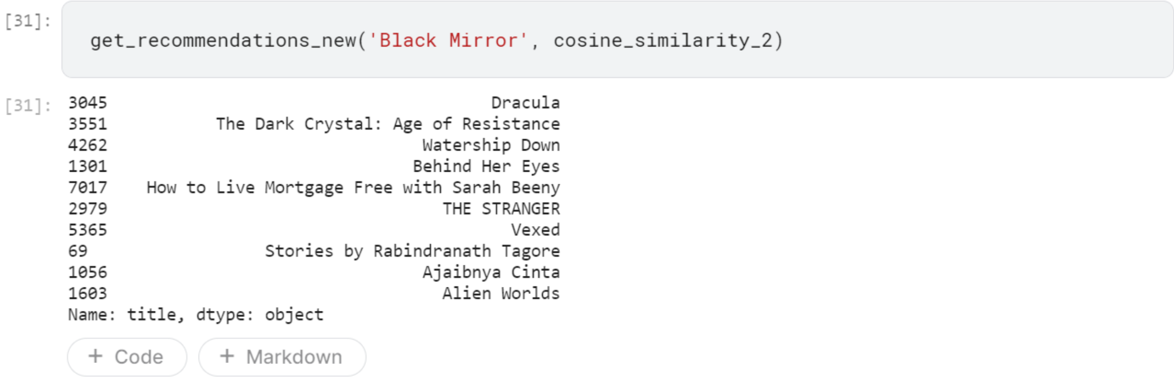
A. Method A

The implementation to construct the system has been described below.

1. Read the input file.
2. Plot graphs and analyze data.
3. Pre-process data.
4. Build a recommendation system using tf-idf and cosine similarity index. Initially, construct the tf-idf matrix by fitting and transforming the data.
5. Compute the cosine similarity matrix.
6. Implement a function to get recommendations based on similarity scores. The function initially includes getting the indices of all the titles. Getting the pairwise

similarity score for all the movies based on that movie. Sort the movies based on the similarity score. Get the scores of 10 most similar movies. Get the movie indices. Return the top 10 most similar movies.

7. Test out the implementation of recommendation of content by passing the content name into the function above.
8. In order to get better accurate results, implement the content based filtering method. Initially identify the features on which the model is to be filtered.
9. Use the count vectorizer method and calculate the cosine similarity.
10. Implement a function to get recommendations. The function is implemented in the same method as above.
11. Test out the implementation of recommendation of content by passing the content name into the function above.
12. Here is an example of the output:



```
[31]:  get_recommendations_new('Black Mirror', cosine_similarity_2)
```

```
[31]: 3045                                      Dracula
      3551             The Dark Crystal: Age of Resistance
      4262                                 Watership Down
      1301                                Behind Her Eyes
      7017     How to Live Mortgage Free with Sarah Beeny
      2979                                   THE STRANGER
      5365                                          Vexed
      69                     Stories by Rabindranath Tagore
      1056                                  Ajaibnya Cinta
      1603                                   Alien Worlds
      Name: title, dtype: object
```

B. Method B

The implementation to construct the system has been described below.

1. Specify Hyper Parameters
2. Load the dataset
3. Clean Data
4. Reduce the number of datapoints by removing movies with too less reviews and users who give too less reviews.
5. Convert the DataFrame with just 3 columns into a large matrix with all unique movies on the rows and unique users on the column and the associated ratings in the matrix cell; also return another boolean DataFrame, R, of same size as df that contains, in ith row and jth column, information about whether ith movie was rated by jth user.
6. Read the movie titles from the provided file and put them in a DataFrame.
7. Randomly initialize parameters X and Theta.
8. Normalize Y.
9. Compute the cost as sum of all the squared errors to be used in fmin_cg.
10. Compute gradients and then flatten the parameters to use with fmin_cg.
11. Computes cost and the gradients to be used in our gradient descent algorithm.
12. Compute Gradient Descent to find minimum cost, return the final X and Theta to be used in our algorithm.
13. Calculate the Root Mean Squared Error.
14. Plot the cost and rmse against multiple epochs.

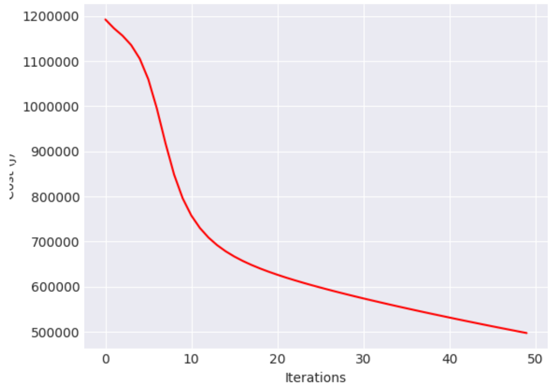
15. Flatten X and Theta to be used in fmincg.
16. Unroll parameters to retrieve X and Theta.
17. Recommend top 10 movies for the user.
18. Calculate RMSE for the probe dataset.

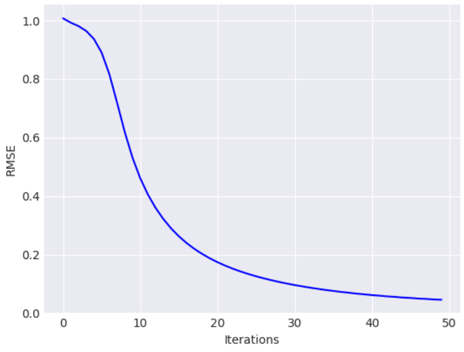
C. **Analysis of Best Performance**.

While implementing the Content Based filtering, based on the count vectorizer method and cosine similarity, the top 10 recommendations for a content were obtained.

While implementing the Collaborative Filtering method, comparison was made between the lowest Cost/RMSE we could get with our gradient descent algorithm with the results generated from passing our cost function and gradients into a highly optimized cost-reduction function called fmincg.

Cost and RMSE calculated on the difference between our y-hat and y in a small dataset of the first 220 movies were both going down.





The gradient descent algorithm:

- Lowest cost: 497385
- Lowest RMSE: 0.04605

For fmincg:

- Lowest cost: 19057
- Lowest RMSE: 0.03714

Section V

Conclusion

While constructing the content-based recommender system, initially a recommender system based on TF-IDF was constructed. But in order to improve the accuracy of recommendations, more metrics are added to the model to improve its performance. Hence, content-based filtering was performed on multiple metrics using the count vectorized method. This particular implementation obtains the top 10 movies based on the content provided.

While constructing the collaborative filtering-based method, the RMSE between yhat and y were was very low, including the cost function went down. This metric was used to evaluate the model's performance.

Though, the dataset used for Content Based Recommender System and Collaborative Filtering based recommendation was different, I would like to mention a few points regarding both the methods.

Advantages of Content Based Recommendation model:

a. No data from other users is required to start making recommendations. Unlike collaborative filtering, content-based filtering doesn't need data from other users to create recommendations.
b. Recommendations are highly relevant to the user.
c. Recommendations are transparent to the user.
d. Easier to create.

Whereas,

Collaborative does not need the features of the items to be given. Every user and item are described by a feature vector or embedding. It creates embedding for both users and items on its own. It also includes implicit feedback and explicit feedback methods which help in better prediction of recommendations. RMSE was used as an evaluation metric here and as we can see, the low score of RMSE and the cost function indicates better results. Future work includes testing this implementation on larger datasets and finding the RMSE and the cost function values. Also, by using other recommender systems on the same dataset and evaluating its performance.

References

1. https://www.analyticsvidhya.com/blog/2015/08/beginners-guide-learn-content-based-recommender-systems/#:~:text=How%20do%20Content%20Based%20Recommender,make%20suggestions%20to%20the%20user.
2. https://developers.google.com/machine-learning/recommendation/content-based/basics
3. https://towardsdatascience.com/introduction-to-recommender-systems-1-971bd274f421
4. https://developers.google.com/machine-learning/recommendation/collaborative/basics
5. https://towardsdatascience.com/intro-to-recommender-system-collaborative-filtering-64a238194a26
6. https://towardsdatascience.com/introduction-to-recommender-systems-1-971bd274f421#:~:text=Content%2Dbased%20filtering%20does%20not,and%20items%20on%20its%20own.
7. Yen-Yao Wang, Andy Luse, Anthony M. Townsend and Brian E. Mennecke, "Understanding the moderating roles of types of recommender systems and products on customer

behavioral intention to use recommender systems", Information Systems and e-Business Management, 2014.

8. Recommender Systems Handbook, 2011.