MOVIE RECOMMENDATION SYSTEM

A PROJECT FOR SURVEY OF MACHINE LEARNING

Contents

Section 01 – Introduction

1. INTRODUCTION

Recommendation systems produce a ranked list of items on which a user might be interested, in the context of his choice of an item. Basically, it provides the user with best possible recommendations.

There are three types of Recommendation system

1. Content based- This type of system is "item" focused. Hence, its recommendations are based on the similarity between the items. For example: recommendation system recommends movies like the selected movie.
2. Collaborative filtering- This type of system is "Item +user" focused. Hence, its recommendations are based on the reaction of similar users like yourself. For example: recommending movies rated high by similar users.
3. Hybrid (Content + Collaborative) - Mix of the above two techniques.

2. PROBLEM STATEMENT

The dataset that we are working on is taken from Kaggle. It consists of movies released on or before July 2017. Data points include cast, crew, plot keywords, budget, revenue, posters, release dates, languages, production companies, countries, TMDB vote counts and vote averages.

This dataset also has files containing 26 million ratings from 270,000 users for all 45,000 movies. Ratings are on a scale of 1-5 and have been obtained from the official GroupLens website.

The problem statement we have is to create a collaborative filtering recommendation system. We intend achieve this goal by discovering the preferences of users from the data, creating a utility matrix consisting of each user-item rating values and predicting the blank cells in this utility matrix.

3. BUSINESS VALUE

a. Why is it important?

We are creating a recommendation system because a good recommendation system plays a very dominant role in the people's choices these days. They drive traffic of the business. for ex: According to Netflix, they earn over a billion in customer retention because the recommendation system accounts for over 80% of the content streamed on the platform.

Converting buyers into consumers takes a unique touch. Individualized communications from a recommendation engine reveal your customer that he is valued as an individual. In turn, this engenders his loyalty.

b. Who in the organization cares?

The volume of data required to create an individual experience for each customer is typically far too huge to be handled manually. Utilizing an engine automates this process, reducing the workload of the IT department without putting a dent financially.

By analyzing the data generated from the recommendation engine, senior management can decide their future set of actions to reach the company's goals.

c. What happens if we solve the problem badly? Who suffers? How can we know if this has happened?

If the recommendation system is not optimal, it can result in producing recommendations in which the user is not interested in and not showing the products the user might like. This can lead to the user feeling that the website does not contain relevant products for his taste.

d. How has this problem traditionally been solved?

Traditionally, recommendation was provided using content-based engines which are pretty one dimensional in their approach and are not personalized.

e. What is in it for us?

We would get some exposure in dealing with Big data and enhance our programming skills.

This project drove us into reading and learning more about recommendation models and tensor flow techniques.

This project has helped us to develop our skills further, broaden our knowledge and expand our portfolio of professional experience.

Section 02 – Project Lifecycle

Collaborative filtering is like recommending a product to your friend who shares similar interests. Using the ratings provided by users for certain products we can cluster these users with similar interests and then recommend these users the items that are rated good by the users of this group.

1. DATA INGESTION & PROCESSING

For collaborating filtering, we need reviews of users on the product that they have invested it. These reviews can either be direct on the numerical scales or can be inform of texts or likes.

Based on this there can be 2 types of data that a business has, and the filtering will be either **Passive** or **Active** filtering.

Since we have direct rating of users for movies on a scale of 1-5. We used **Active filtering** approach.

The data is converted to matrix form where each column represents user and rows represent item (in our case, movies) and each cell represent the rating that a user "j" gave to movie "i".

We do not attempt to fill NA values as we do in other modeling techniques. One of the reasons is that we are trying to address those NA values as a result of the collaborative filtering model. The idea is that NA value represents the product which is not reviewed by user. We make this **assumption** that if the user has not reviewed it, he most likely has not watched it. So, we can predict what rating the user might give to the item. If the rating is above "3" we would recommend this to the user.

2. MODEL DEVELOPMENT

As mentioned above we will be working on collaborative filtering model. The approach that is taken for this model building is we divide the matrix of user-item rating into 2 matrices. This is also called matrix factorization[1]

Let us say we have m users rated n movies. So, we have a m x n matrix. The idea is that we have f features based on what users rate the item and x features that an item has. So, we divide the m x n matrix into 2 matrices of m x f and f x n dimensions.

We then represent each of these ratings as a dot product of these matrices. So, our cost function comes out to be the sum of the squared differences of the predicted rating and the actual rating.

There are several ways of performing matrix factorization:

1. Low Rank Matrix Factorization using Least Square Methods
2. SVD – Single Value Decomposition
3. PCA

[1] https://datajobs.com/data-science-repo/Recommender-Systems-[Netflix].pdf

For our project we decided to go with Low Rank Matrix Factorization approach as it gives us more granular control over how to treat our NA values. We used Adam's optimization algorithm for minimizing this cost function.

3. MODEL EVALUATION

Model evaluation methods for recommendation systems are same as what we do for other prescriptive models. One of the most common and efficient way is to collect user feedback. This can be done either on the frequency basis or with every recommendation that we make, we could ask user if they liked this recommendation.

However, this has a problem as not every user provides the feedback unless our recommendation is too good or too bad. To mitigate this problem, we could track the user activity for an item after we have recommended it to him.

Activities like whether user completely watched that movie. He started watching the movie. He added the movie to his watch list. We can then assign weights to these activities and come up with a metric of whether user liked the recommendation or not.

Section 03 – Results and Conclusion

1. RESULTS

Once we run our model on the current data, we get an estimate of rating that a user will provide to a movie that he has not watched. Now, we sort these ratings in the descending order and pick first K movies to recommend to this user, where K would be the number of movies we want to recommend.

For illustration purposes we will select user with id 1 (the name of the user is not provided for privacy concerns)

Top Rated Movies

original_title	genres
Sleepers	[{'id': 80, 'name': 'Crime'}, {'id': 18, 'name': 'Drama'}, {'id': 53, 'name': 'Thriller'}]
Nuovo Cinema Paradiso	[{'id': 18, 'name': 'Drama'}, {'id': 10749, 'name': 'Romance'}]
The French Connection	[{'id': 28, 'name': 'Action'}, {'id': 80, 'name': 'Crime'}, {'id': 53, 'name': 'Thriller'}]
Tron	[{'id': 878, 'name': 'Science Fiction'}, {'id': 28, 'name': 'Action'}, {'id': 12, 'name': 'Adventure'}]
Blazing Saddles	[{'id': 37, 'name': 'Western'}, {'id': 35, 'name': 'Comedy'}]

Recommended Movies

original_title	genres
Pulp Fiction	[{'id': 53, 'name': 'Thriller'}, {'id': 80, 'name': 'Crime'}]
The Shawshank Redemption	[{'id': 18, 'name': 'Drama'}, {'id': 80, 'name': 'Crime'}]
Schindler's List	[{'id': 18, 'name': 'Drama'}, {'id': 36, 'name': 'History'}, {'id': 10752, 'name': 'War'}]
One Flew Over the Cuckoo's Nest	[{'id': 18, 'name': 'Drama'}]
The Matrix	[{'id': 28, 'name': 'Action'}, {'id': 878, 'name': 'Science Fiction'}]

2. CONCLUSION

In today's business scenario where customer value comprises more than just the product or services that is given to him, recommendation systems brings in immense benefit. It considers, customer's personal tastes and preferences and makes customer feel valued.

The collaborative filtering approach suffers from "cold-start" problem. It means that at the starting of the business when we do not have much the data/ratings to begin with, there is no way we can make it work. As mentioned, it mimics the "word to mouth recommendation", we cannot recommend a product to anyone if no one has tried it ever. Therefore, it is best used when combined with content-based recommendation system. This together is known as "**Hybrid Recommendation System".** However, this system alone works pretty-well too if applied to already established business.

If an organization is evaluating the idea of integrating a recommendation system to their existing business, collaborative filtering comes with fastest and cheapest implementation, as it does not require complex data collection. Company works with the data they already have and all they need is user, item and rating.

Section 04 – References

1. http://www.zheng-wen.com/WenRecommendation.pdf
2. https://datajobs.com/data-science-repo/Recommender-Systems-[Netflix].pdf
3. https://datajobs.com/data-science-repo/Recommender-Systems-[Netflix].pdf
4. https://www.themoviedb.org/documentation/api
5. Stanford's Machine Learning course.