Book Recommendation System

Nikunj Sonule
Data Science Trainee
Almabetter, Bangalore

Abstract:

A recommendation engine is a class of machine learning which offers relevant suggestions to the customer. A recommendation system is one of the top applications of data science. Every consumer Internet company requires a recommendation system like Netflix, YouTube, a news feed, etc. What you want to show out of a huge range of items is a recommendation system. Before the recommendation system, the major tendency to buy was to take a suggestion from friends. But Now Google knows what news you will read, YouTube knows what type of videos you will watch based on your search history, watch history, or purchase history. We were provided with 'Book Recommendation System' dataset to perform machine learning task, were to get insights from users-items interactions and provide best recommendation to users.

Our experiment can help understand what could be the reason for unsupervised problems of such labels by feature engineering, data analysis and prediction with machine learning models taking into account previous trends to determine unsupervised problems.

Keywords:machine learning, unsupervised, collaborative filtering, recommendation system

1. Problem Statement

During the last few decades, with the rise of YouTube, Amazon, Netflix, and many other such web services, recommender systems have taken more and more place in our lives. From e-commerce (suggest to buyers articles that could interest them) to online advertisement (suggest to users the right contents, matching their preferences), recommender systems are today unavoidable in our daily online journeys. By analyzing the problems with 'Book Recommendation System' feature, how we can predict the best recommendation for users according to their items approach.

A recommendation system helps an organization to create loyal customers and build trust by them desired products and services for which they came on your site. The recommendation system today is so powerful that they can handle the new customer too who has visited the site for the first time. They recommend the products which are currently trending or highly rated and they can also recommend the products which bring maximum profit to the company.

In a very general way, recommender systems are algorithms aimed at suggesting relevant items to users (items being movies to watch, text to read, products to buy, or anything else depending on industries). Recommender systems are really critical in some industries as they can generate a huge amount of income when they are efficient or also be a way to stand out significantly from competitors. The main objective is to

create a book recommendation system for users.

We've all record and data with three different dataset – Book dataset (ISBN, Book-Title, Book-Author, Year-Of-Publication, Publisher, Image-URL-S, Image-URL-M, Image-URL-L); Users dataset (User-ID, Location, Age); Ratings dataset (User-ID, ISBN, Book-Rating). Providing specific data analysis and prediction to done with this data. The main objective is to built a predictive recommender model, which could help in predicting – how we can predict the best recommendation for users according to their items approach. This would help us in providing better recommendation item to a right specific users.

- Book data – (ISBN, Book-Title, Book-Author, Year-Of-Publication, Publisher, Image-URL-S, Image-URL-M, Image-URL-L);
- Users data - (User-ID, Location, Age);
- Ratings data - (User-ID, ISBN, Book-Rating)

Our goal is to build a predictive recommendation model, which could help company in predicting get insights from users-items interactions and provide best recommendation to users.

2. Introduction

A book recommendation system is a type of recommendation system where we have to recommend similar books to the reader based on his interest. The books recommendation system is used by online websites which provide ebooks like google play books, open library, good Read's, Amazon Kindle etc. We've to use the Collaborative based filtering method to build a book recommender system. The major concern is about providing best recommendation to users. Looking towards various factor/feature we can take leverage of data to make better predictions on testset. To let proceed with recommendation it's important to have every users interactions with items.

Generally, the models can predict best recommendation for the data according to it RMSE score. As for Collaborative Filtering we'll be focusing on RMSE. This recommendation system data can show how it varies from every machine learning approach. So, generating different RMSE score from different method can reveal best recommendation.

3. Users-Items description 0f data

The users interaction is very vital role for recommendation. To successful build collaborative filtering model in recommender system, data preparation is important. Beginning with book data – dropping URL features (i.e 'Image-URL-S', 'Image-URL-M', 'Image-URL-L'). We have some extra columns which are not required for our task like image URLs. And we rename the columns of each file as the name of the column contains space and lowercase letters so we will correct as to make it easy to use. The features depict great analysis by feature engineering.

4. Observation from features and hypothetical assumption

The dataset is reliable and can consider as a large dataset. We have 271360 books data and total registered users on the website are approximately 278000 and they have given near about 11 lakh rating. hence we can say that the dataset we have is nice and reliable.

Agatha Christie is leading at top with more than 600 counts, followed by William Shakespeare. It can happen in some possible cases that Agatha Christie is not a best Author, though Agatha Christie has most number of books as compared to others. William Shakespeare is one of the popular Author in the world. Still he doesn't have highest number of books. Among all other Authors, it might happen that few of the Author might have some of the best seller books who have millions of copies been sold in world.

Harlequin has most number of books published, followed by Silhouette. Hypothetical assumptions - Some of the top Author's had published their books from Harlequin. We can observe Harlequin publisher's marking better performance than any other publishers. Penguin Books, Warner Books, Penguin USA, Berkely Publishing Group and many more are among popular publisher's remarking competition with Harlequin. Though Penguin Books Publisher has less number of books published but it might happen that only top Author's are approaching towards Penguin Books Publisher.

There are 4618 entries as ‘0’ and 0 NaN entries in the Year of Publication field. Publication years are somewhat between 1950 - 2005.The publication of books got vital when it starts emerging from 1950. It might happen people starts to understand the importance of books and gradually got productivity habits in their life. Every user has their own taste to read books based on what particular subject Author uses. The subject of writing books got emerge from late 1940 slowly. Till 1970 it has got the opportunity to recommend books to people or users what they love to read. The highest peak we can observe is between 1995-2001 year. The user understands what they like to read. Looking towards the raise the recommendation is also increase to understand their interest.

Looking towards the users age between 30-40 prefer more and somewhat we can also view between 20-30. It is obvious that most of the user books are from Age 30 to 40. It might happen that the users are more interested on that subject what Authors are publishing in the market. The age group between 20-30 are immensely attracted to read books published by Author. We can observe same pitch for Age group between 10-20 and 50-60. There can be lot of different reasons.

As per ratings "Selected Poems" has been rated most followed by "Little Women". Selected Poems are most favorable to users as per ratings. Three of the books 'The Secret Garden', 'Dracula', 'Adventures of Huckleberry Finn' are struggling to compete with each other. Similarly, we can observe in 'Masquerade', 'Black Beauty', 'Frankenstein'. Firstly the above ratings are unique ratings from 'ratings_data' and 'books_data' dataset.
We have separate the explicit ratings represented by 1–10 and implicit ratings represented by 0. Let's make some

hypothesis assumptions - Mostly the users have rated 8 ratings out of 10 as per books. (i.e best books ever). Now this countplot of bookRating indicates that higher ratings are more common amongst users and rating 8 has been rated highest number of times. There can be many assumptions based on ratings of users - taking ratings group from 1-4. This can be negative impact for books been published if they have ratings from 1 to 4. It can be issues related to language, offend by any chapter's incident/paragraph/Author, they've read worst book ever. For 5 ratings the users might not sure about book ratings whether it's positive or negative impact. take ratings group from 6-10. This are positive feedback - It can happen that not every book is perfect in all desire. So, the user's have decided to rate 8. Since 6 ratings is very low among other ratings. As we can aspect 7 and 8 are average and more ratings from users. 9 and 10 ratings are top best ratings based on Author's, Publisher's and Books been published.

5. Top rated books as per ratings

Building recommendation system based on popularity (i.e ratings). These recommendations are usually given to every user irrespective of personal characterization. Merged book_data dataset and ratings_explicit. Considering ISBNs that were explicitly rated for this recommendation system. So we achieved top ten books as per ratings –

book_title	book_rating
The Lovely Bones : A Novel	707
Wild Animus	581
The Da Vinci Code	494
The Secret Life of Bees	406
The Nanny Diaries : A Novel	393
The Red Tent (Bestselling Backlist)	383
Bridget Jones's Diary	377
A Painted House	366
Life of Pie	336
Harry Potter and the Chamber of Secrets (Book 2)	326

It might happen that the feedback is positive but not extremely positive as 10 ratings

The above are the top 10 books recommendation as per ratings. But this are not based on some recommendation system. They are top 10 books as per ratings.

6. Steps Involved

- **Exploratory Data Analysis**

Analytics for every dataset (i.e book, users, ratings) has helped to understand user-item interactions for book recommendation. Viewed top books as per ratings. Analysis based on top authors with highest number of books, top publishers with highest number of books, number of books published in yearly, users age distributions, top books as per ratings, different various user's ratings.

- **Null values treatment**

We served with three dataset (i.e book_dataset, users_dataset, ratings_dataset). We got null values in book dataset (features are book-author, publisher, Image-URL-L), users dataset in age. Ratings dataset doesn't contain any null values.

- **Dropping and replacing data**

Proceeding with data cleaning and feature selection is a crucial steps – we dropped feature like image URL. Replaced feature with lowercase and '-' to built a space between words. Some of the null values were present in feature data, we replaced with mean of that particular feature. Deal with mismatch feature like book_title, book_author, year_of_publication, publisher. Only considering age between 5-90 we took users data to analysis and perform recommendation on it.

- **Univariate Analysis**

Data Visualization is a major part of project to understand each perspective view. We perform analysis on single feature. It could be simple and easy stage but more effectively to focus. Both numeric and categorical feature has helped with analysis and got insight. Each feature like book_title, isbn, book_author, year_of_publication, publisher, user_id, age, book_rating etc. shows great impact on analysis and training the recommendation model.

- **Collaborative Filtering in memory based and model based**

Collaborative Filtering (CF) techniques make collaborative research and process over user or item ratings to deduce new recommendations for users. This collaborative research includes finding similarities between users and items to make assumptions for missing rating values and deducing new recommendations.

Memory based approach was our first trial on train and test dataset which uses the memory of previous users interactions to compute users similarities based on items they've interacted (i.e user-based approach) or compute items similarities based on the users that have interacted with them (i.e item-based approach). Applying cosine similarity to make item-item similarity need to take transpose of matrix. This matrix would help in manage train-test matrix. After all views predictions based on similarity, we find recommendation on it based on score.

Model-based collaborative filtering algorithms provide item recommendation by first developing a model of user ratings. Algorithms in this category take a probabilistic approach and envision the collaborative filtering process as computing the expected value of a user prediction, given his/her ratings on other items. We use Latent Factor Model called Singular Value Decomposition (SVD). SVD made dimensionality reduction technique in machine learning. SVD is a matrix factorization technique, which reduces the number of features of a dataset by reducing the space dimension from N-dimension to K-dimension (

where K<N). It uses a matrix structure where each row represents a user, and each column represents an item. The elements of this matrix are the ratings that are given to items by users.

- **Model Evaluation Metrics**

Model evaluation metrics is important to distinguish the best collaborative filtering – either by memory based or model based approach. The memory based approach – Cosine Similarity shows RMSE score for item based CF is 8.00 and for user based CF it shows 8.00. The score is slightly similar. Model based collaborative filtering made it better score with Latent Factor Model called SVD. The score improved to 1.63 for both SVD RMSE and accuracy score.

7. Collaborative Filtering

Collaborative Filtering (CF) techniques make collaborative research and process over user or item ratings to deduce new recommendations for users. This collaborative research includes finding similarities between users and items to make assumptions for missing rating values and deducing new recommendations. CF techniques are grouped in two methods: Memory-based and Model-based methods.

- **Memory based CF**

Memory-Based Collaborative Filtering approaches can be divided into two main sections: user-item filtering and item-item filtering. A user-item filtering takes a particular user, find users that are similar to that user based on similarity of ratings, and recommend items that those similar users liked. In contrast, item-item filtering will take an item, find users who liked that item, and find other items that those users or similar users also liked. It takes items and outputs other items as recommendations.
Item-Item Collaborative Filtering: "Users who liked this item also liked …"
User-Item Collaborative Filtering: "Users who are similar to you also liked …"

The key difference of memory-based approach from the model-based techniques is that we are not learning any parameter using gradient descent (or any other optimization algorithm). The closest user or items are calculated only by using Cosine similarity.

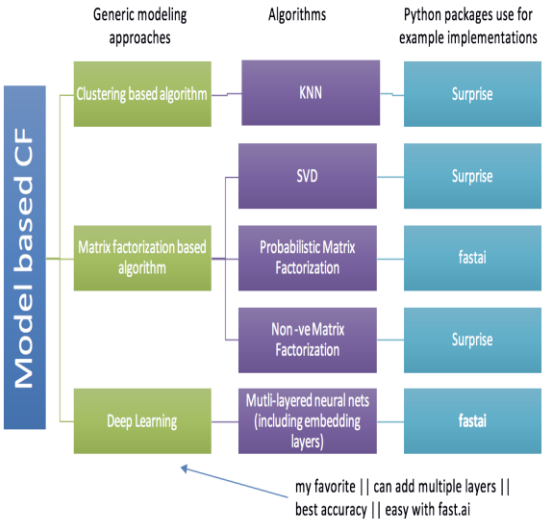
A common distance metric is cosine similarity. For user-based collaborative filtering, two users' similarity is measured as the cosine of the angle between the two users' vectors. For users u and u′, the cosine similarity is:

$$sim(u,u') = cos(\theta) = \frac{\mathbf{r}_u \cdot \mathbf{r}_{u'}}{\|\mathbf{r}_u\|\|\mathbf{r}_{u'}\|} = \sum_i \frac{r_{ui} r_{u'i}}{\sqrt{\sum_i r_{ui}^2}\sqrt{\sum_i r_{u'i}^2}}$$

As no training or optimization is involved, it is an easy to use approach. But its performance decreases when we have sparse data which hinders scalability of this approach for most of the real-world problems.

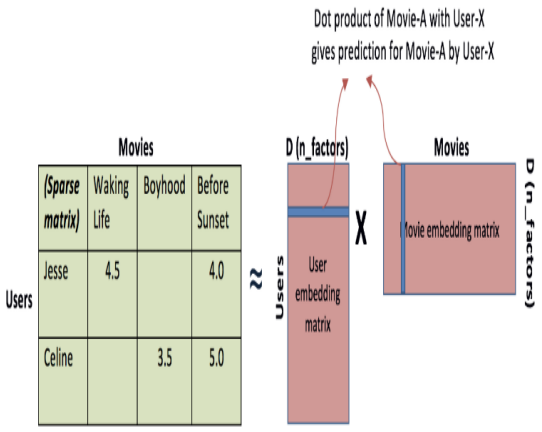
- **Model based CF**

In this approach, CF models are developed using machine learning algorithms to predict user's rating of unrated items. As per my understanding, the algorithms in this approach can further be broken down into three sub-types.



- Matrix Factorization (MF): The idea behind such models is that attitudes or preferences of a user can be determined by a small number of hidden factors. We can call these factors as Embeddings.

Intuitively, we can understand embeddings as low dimensional hidden factors for items and users.



In the SVD model, an estimated rating of user u on item i is calculated as:

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T p_u$$

where μ is the overall average rating, and every other parameter is calculated from the model with a gradient descent method. So the model will try to fit this estimated rating on all the known ratings, minimise the MSE, and return the closest fit. b_u and b_i are scalars, they represent the biases of the user u or item i.

p_u and q_i are vectors, and their length is a hyperparameter of the model, n. They are the actual matrix-factorisation part of the model, that is where the magic happens. Each user and item will be represented by their vector, that tries to capture their essence in n numbers. And we get the rating by multiplying the item — user pairs (and adding averages and biases of course.

Training SVD model like other models and test the model performance using RMSE score (which stands for Root Mean Squared Error, the lower the better).

- **Model Evaluation metrics**

RMSE score is a best way to model performance for recommendation system. Root Mean Square Error (RMSE) is a standard way to measure the error of a model in predicting quantitative data.

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

Model evaluation metrics is important to distinguish the best collaborative filtering – either by memory based or model based approach. The memory based approach – Cosine Similarity shows RMSE score for item based CF is 8.00 and for user based CF it shows 8.00. The score is slightly similar. Model based collaborative filtering made it better score with Latent Factor Model called SVD. The score improved to 1.63 for both SVD RMSE and accuracy score.

8. Conclusion

That's it! I reached the end of exercise. Starting with loading the data so far I have done data cleaning and feature engineering, null values treatment, some univariate analysis. Collaborative Filtering was among best method to approach recommendation system for this project. Model based approach like Latent Factor Model called SVD and Memory based approach with cosine similarity was model building approach. The comparison of RMSE score between model and memory based approach was quite different. Model Evaluation metrics shows better recommendation with model based CF. The RMSE score varies in both the model but optimal model we can find is in SVD. So SVD with RMSE score is the best model with 1.63 for this dataset. This performance could be due to various reasons : pattern of data, different model give different accuracy score, business understanding, machine learning approach etc. Finally Singular Value Decomposition (SVD) is an optimal model for book recommendation system of this dataset.

References:-

1. towardsdatascience
2. indiaai.gov.in
3. analyticsvidhya