CAPSTONE PROJECT – BOOK RECOMMENDATION SYSTEM

MANAS RANJAN BEHERA
DATA SCIENCE TRAINEE, BANGALORE

Abstract:

Recommendation systems is used for the purpose of suggesting items to purchase or to see. They direct users towards those items which can meet their needs through cutting down large database of Information. A various techniques have been introduced for recommending items i.e. content, collaborative and hybrid techniques are used. This paper solves the problem of data sparsity problem by combining the collaborative-based filtering and SVD(Single Value Decomposition) to achieve better performance and the trends are achieved by principal of popularity. The results obtained are demonstrated and the proposed recommendation algorithms perform better and solve the challenges such as data sparsity and understanding the metric evaluation.

INTRODUCTION:

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. 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). Personal recommendation systems have been emerged to conduct effective search which related books based on user rating and interest. The proposed system used the K-NN K-NN Cosine Distance function to measure distance and Cosine Similarity function to find Similarity between the books clusters also we implemented SVD system that give us good recommendation.

OBJECTIVE:

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 objective of the project is to build a book recommendation system for users based on popularity and user interests. The next chapters have the following sections which are the steps involved for solving the Book Recommendation System,

Section 1 - Data collection

Section 2 - Data preparation

Section 3 - Exploratory data analysis

Section 4 - Feature Engineering

Section 5 - Working different models

Section 6 - Evaluating model

DATA COLLECTION AND DATA PREPARATION

1 Data Summary:

We are using Book-Crossing dataset to train and test our recommendation system. Book-Crossings is a book ratings dataset compiled by Cai-Nicolas Ziegler. It contains 1.1 million ratings of 270,000 books by 90,000 users. The ratings are on a scale from 1 to 10. The Book-Crossing dataset comprises 3 files.

● **Users**

This .csv file contains the users. Note that user IDs (User-ID) have been anonymized and map to integers. Demographic data is provided (Location, Age) if available. Otherwise, these fields contain NULL values.

● **Books**

Books are identified by their respective ISBN. Invalid ISBNs have already been removed from the dataset. Moreover, some content-based information is given (BookTitle, Book-Author, Year-Of-Publication, Publisher), obtained from Amazon Web Services. Note that in the case of several authors, only the first is provided. URLs linking to cover images are also given, appearing in three different flavors (Image-URLS, Image-URL-M, Image-URL-L), i.e., small, medium, large. These URLs point to the Amazon website.

● **Ratings**

Contains the book rating information. Ratings (Book-Rating) are either explicit, expressed on a scale from 1-10 (higher values denoting higher appreciation), or implicit, expressed by 0

2 Data Collection

Before building any machine learning model, it is vital to understand what the data is, and what are we trying to achieve. Data exploration reveals the hidden trends and insights and data preprocessing makes the data ready for use by ML algorithms.

So, let's begin. . .

To proceed with the problem dealing first we loaded our dataset that is given to us in a .csv file into a data frame. Mount the drive and load the csv file into a data frame.

The primary goal of EDA is to support the analysis of data prior to making any conclusions. Exploratory data analysis is an approach of analysing data sets to summarize their main characteristics, often using statistical graphics and other data visualization methods.

Books_dataset

In the books dataset we have the following feature variables.

● ISBN (unique for each book)

● Book-Title

● Book-Author

● Year-Of-Publication

● Publisher

● Image-URL-S

● Image-URL-M

● Image-URL-L

We found the top 10 Book-Author and top 10 Books. Further we found that both the plots are skewed and the maximum number of books are from top 10 Book-Authors and top 10 Books.

From bar plot we observed that Agatha Christie wrote highest number of books in our given dataset selected poems book is ranked one in the list of top 10 books.

Users_Dataset

In the users_dataset we have the following feature variables.

● User-ID (unique for each user)

● Location (contains city, state and country separated by commas)

● Age

Out of these features, User-ID is unique for each user, Location contains city, state and country separated by commas and we have Age given across each user.

We observed that 41.9% of age 34 group read more books compared to other age groups. Also the users with the age 60 and above do not read more books.

We observed that user with locations London, England, United Kingdom, Toronto, Ontario, Canada are high in numbers.

Ratings Dataset:

In the ratings_dataset we have the following feature variables.

● User-ID

● ISBN

● Book-Rating

Here we found that the ratings are very unevenly distributed, and the vast majority of ratings are 0 .As mentioned in the description of the dataset - BX-Book-Ratings contains the book rating information. Ratings are either explicit, expressed on a scale from 1-10 higher values or implicit, expressed by 0. Hence segregating implicit and explicit ratings dataset.

It can be observed that higher ratings are more common amongst users and rating 8 has been rated the highest number of time.

Data Cleaning:

There were missing values in some columns in our dataset.

- The book dataset ‘year’ column have 4690 missing values.
- The user dataset ‘Age’ column have 40% missing values.

In the data cleaning section we will drop off the features which do not contribute towards making a good recommendation system.

We have already seen in our EDA part that the Image-URL-S, Image-URL-M and Image-URL-L do not contribute towards our final goal. So, we dropped off these columns and also dropped these columns we will not be losing any information. We renamed the columns of each file. Because the name of the column contains space, and uppercase letters so we will correct as to make it easy to use.

<u>FEATURE ENGINEERING</u>

Handling missing values:

Missing data present various problems. First, the absence of data reduces statistical power, which refers to the probability that the test will reject the null hypothesis when it is false. Second, the lost data can cause bias in the estimation of parameters. Third, it can reduce the representativeness of the samples. Fourth, it may complicate the analysis of the study. Each of these distortions may threaten the validity of the trials and can lead to invalid conclusions. Here in our case we had a lot of data points which were having missing values in different feature columns.

1 Imputing Values:

Since, we have missing values in some feature variables, we imputed them.

● Year-Of-Publication

● Book-Author

● Publisher

● Age

Year-Of-Publication:

As it can be seen from below that there are some incorrect entries in this field. It looks like Publisher names 'DK Publishing Inc' and 'Gallimard' have been incorrectly loaded as year in dataset due to some errors in csv file. Also some of the entries are strings and same years have been entered as numbers in some places.

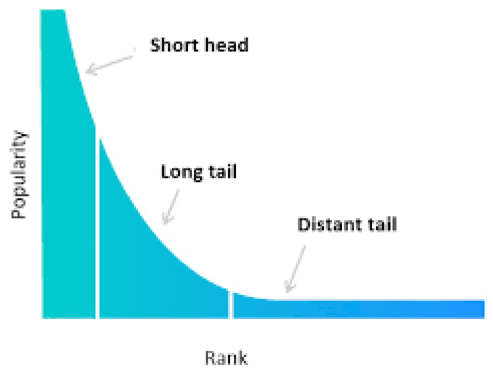
From above data frame we can see that there has been a shift in the values of this three rows. We brought them in the right place.

Age:

Age column has some invalid entries like nan, 0 and very high values like 100 and above. In our view values below 5 and above 90 do not make much sense for our book rating case...hence replacing these by Nan’s.

DIFFERENT MODELS

1 . POPULARITY BASED RECOMMEND-ATION



It is a type of recommendation system which works on the principle of popularity and or anything which is in trend. These systems check about the books which are in trend or are most popular among the users and directly recommend them.

For example, if a product is often purchased by most people then the system will get to know that that product is most popular so for every new user who just signed it, the system will recommend that product to that user also and chances become high that the new user will also purchase that.

Merits of Popular based recommendation system:

The popularity-based recommendda-tion system eliminates the need for knowing other factors like user browsing history, user preferences, genre, and other factors. Hence, the single-most factor considered is the star rating to generate a scalable recommendation system. This increases the chances of user engagement as compared to when there was no recommendation system.

Cold start is a potential problem in computer-based information systems which involves a degree of automated data model-ling. Specifically, it concerns the issue that the system cannot draw any inferences for users or items about which it has not yet gathered sufficient information. There are three cases of cold start:

• ***New community:*** refers to the start-up of the recommender, when, although a catalogue of items might exist, almost no users are present and the lack of user interaction makes it very hard to provide reliable recommendations.

• ***New item:*** a new item is added to the system, it might have some content information but no interactions are present.

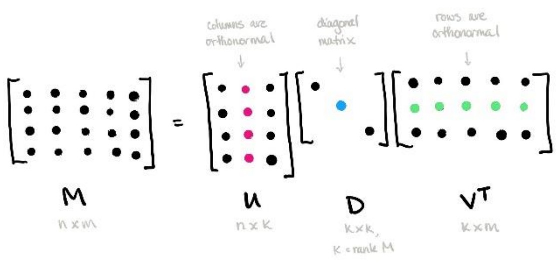
• ***New user***: a new user registers and has not provided any interaction yet, therefore it is not possible to provide personalized recommendations.

A popularity based model does not suffer from cold start problems which means on day 1 of the business also it can recommend products on various different filters. There is no need for the user's historical data.

Now let's try to build our first recommendation system based on popularity. These systems check about the product which are in trend or are most popular among the users and directly recommend those.

2 . MODEL-BASED RECOMMENDATION

Singular Value Decomposition (SVD):



Singular value decomposition also known as the SVD algorithm is used as a collaborative filtering method in recommendation systems. SVD is a matrix factorization method that is used to reduce the features in the data by reducing the dimensions from N to K where (K<N).

For the part of the recommendation, the only part which is taken care of is matrix factorization that is done with the user-item rating matrix. Matrix-factorization is all about taking 2 matrices whose product is the original matrix. Vectors are used to represent item ‘qi’ and user ‘pu’ such that their dot product is the expected rating as given below

$$\text{expected rating} = \hat{r}_{ui} = q_i^T p_u$$

qi’ and ‘pu’ can be calculated in such a way that the square error difference between the dot product of user and item and the original ratings in the user-item matrix is least.

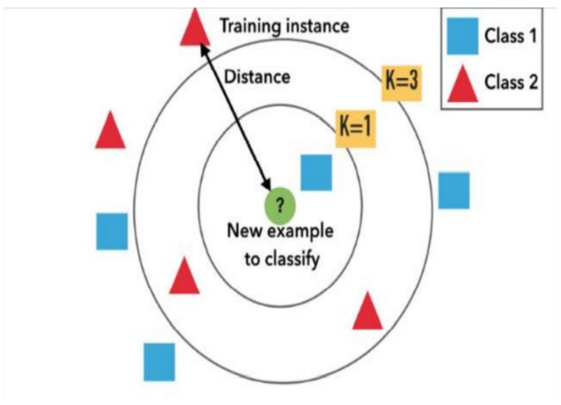
Benefits of using SVD?

There are 3 primary benefits : ● It’s very efficient ● The basis is hierarchical, ordered by relevance ● It tends to perform quite well for most data sets Surprise is a Python scikit for building and analysing recommender systems that deal with explicit Rating data. The name SurPRISE (roughly) stands for Simple Python Recommendation System Engine

3 . COLLABORATIVE FILTERING (ITEM-ITEM-BASED)

I used K-Nearest Neighbour and Cosine Similarity.

kNN (k-Nearest Neighbors) as an algorithm seems to be inspired from real life. The full k-Nearest Neighbours algorithm works much in the way some of us ask for recommendations from our friends. First, we start with people whose taste we feel we share, and then we ask a bunch of them to recommend something to us. If many of them recommend the same thing, we deduce that we’ll like it as well. Our behaviour is guided by the friends we grew up with kNN is a machine learning algorithm to find clusters of similar users based on common book ratings, and make predictions using the average rating of top-k nearest neighbours. KNN does not make any assumptions on the underlying data distribution but it relies on item feature similarity. When KNN makes inference about a movie, KNN will calculate the “distance” between the target book and every other book in its database, then it ranks its distances and returns the top K nearest neighbor movies as the most similar book recommendations.



Here we assume that users who given ratings more than 200 are users who read at least 20 books (suppose on user given rating 10/10 so minimum he read books (200 ratings/10 ratings per book=20).Hence for statistical significance we should consider only the data of user who given more than 200 ratings.

First we will the dataset 'Ratings' and 'Books' have common column 'ISBN' so create new data frame merging the two data frames then we will group by book titles and create a new column for total rating count.

This gives us exactly what we need to find out which books are popular and filter out lesser-known books. We will consider the only books having minimum total 50 ratings.

Wait, but how do we feed the data frame of ratings into a KNN model? First, we need to transform the data frame of ratings into a proper format that can be consumed by a KNN model. We want the data to be in an m x n array, where m is the number of books and n is the number of users. To reshape data frame of ratings, we'll pivot the data frame to the wide format with books as rows and users as columns. Then we'll fill the missing observations with 0s since we're going to be performing linear algebra operations (calculating distances between vectors). Let's call this new data frame a data frame of books features.

Now we can fit our model with pivot matrix. We used metric cosine and algorithm brute.

4 . COLLABORATIVE FILTERING (USER-ITEM-BASED)

User-Based Collaborative Filtering is a technique used to predict the items that a user might like on the basis of ratings given to that item by the other users who have similar taste with that of the target user. Many websites use collaborative filtering for building their recommendation system.

Steps for User-Based Collaborative Filtering:

Step 1: Finding the similarity of users to the target user U. Similarity for any two users 'a' and 'b' can be calculated from the given formula,

$$Sim(a,b) = \frac{\sum_p (r_{ap}-\bar{r}_a)(r_{ab}-\bar{r}_b)}{\sqrt{\sum(r_{ap}-\bar{r}_a)^2}\sqrt{\sum(r_{bp}-\bar{r}_b)^2}}$$

$$r_{up} : rating\ of\ user\ u\ against\ item\ p$$

$$p : items$$

Step 2: Prediction of missing rating of an item Now, the target user might be very similar to some users and may not be much similar to the others. Hence, the ratings given to a particular item by the more similar users should be given more weightage than those given by less similar users and so on. This problem can be solved by using a weighted average approach. In this approach, you multiply the rating of each user with a similarity factor calculated using the above mention formula. The missing rating can be calculated as,

$$r_{up} = \bar{r}_u + \frac{\sum_{i \in users} sim(u,i) * r_{ip}}{\sum_{i \in users} |sim(u,i)|}$$

CONCLUSION

In EDA, the Top-10 most rated books were essentially novels. Books like The Lovely Bone and The Secret Life of Bees were very well perceived.

Majority of the readers were of the age bracket 20-35 and most of them came from North American and European countries namely USA, Canada, UK, Germany and Spain.

If we look at the ratings distribution, most of the books have high ratings with maximum books being rated 8. Ratings below 5 are few in number.

Author with the most books was Agatha Christie, William Shakespeare and Stephen King.

For modelling, it was observed that for model based collaborative filtering SVD technique worked way better than NMF with lower Mean Absolute Error (MAE) .

CHALLENGES

Handling of sparsity was a major challenge as well since the user interactions were not present for the majority of the books.

Understanding the metric for evaluation was a challenge as well.

Since the data consisted of text data, data cleaning was a major challenge in features like Location etc..

Decision making on missing value imputations and outlier treatment was quite challenging as well.