INDIAN INSTITUTE
OF TECHNOLOGY
PALAKKAD

A Machine Learning Lab Project report on

The books recommendation system

Submitted by

Ajay Kumar Pal

Chitresh Kulhade

Under the guidance of

Dr. Sahely Bhadra

ACKNOWLEDGEMENT

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them. We are highly indebted to Dr. Sahely Bhadra for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

We would like to express our special gratitude and thanks to Teaching Assistance of Data Science Department for giving us such attention and time.

Our thanks and appreciations also go to our colleague in developing the project and people who have willingly helped us out with their abilities.

ABSTRACT

A consumer based company totally depends upon how much time a user is spending on their platform while consuming their product. But user have a variety of platforms and also a lot of content within a particular platform to choose from, which can be time consuming and tedious.

Hence every consumer based company requires a recommendation system which maintains a constant interest within user and also saves his/her valuable time.

A recommendation system is a machine learning algorithm which offers relevant suggestions to the user. Our Book Recommendation system helps the platforms to create loyal customers by building trust between customers and organisation by providing the best relevant content available in their platform.

A recommendation system makes prediction based on users' historical behaviour. We have used two types of recommendation systems, first one is based on collaborative filtering and second one is content based filtering technique.

Collaborative filtering helps in recommending the books based on highest rating given by other users, where as content based filtering helps in suggesting the books based on the type of content a user is interested in.

To compensate the limitations of above mentioned methods, two other methods based on popularity and weighted average also have been discussed.

Collaborative-Based Filtering :-

It is a machine learning algorithm which considers data collected by the system, along with the interaction of the users with the data in the past. The basic idea is that if a person likes a certain book then user similar to that person will also like that book.

For example, we often asks a friend or colleague to recommend a book whenever we want to read a new book. This is because we trust the recommended book as it was suggested by a friend/colleague who shares similar taste in books.

In this recommendation system, we have used the neighbourhood based approach, i.e. numbers of users are considered based on their similarity to the target user. The similarity between two users will be based upon the historical data i.e. similarity between ratings given by both users to a certain book.

In user based collaborative filtering,

we have n*m matrix of ratings, with user u_i , i = ,..., n

book p_j , j = ,..., m.

If target user ' i ' did not read book ' j ', then predicting the ratings r_{ij}

This process involves the calculation of the similarities between target user ' i ' and all the other users.

$$r_{ij} = \frac{\sum_k Similaries(u_i, u_k) r_{kj}}{number\ of\ ratings}$$

Similarity is calculated by **Cosine similarity**

$$Cosine\ Similarity : Sim(u_i, u_k) = \frac{r_i \cdot r_k}{|r_i||r_k|} = \frac{\sum_{j=1}^{m} r_{ij} r_{kj}}{\sqrt{\sum_{j=1}^{m} r_{ij}^2 \sum_{j=1}^{m} r_{kj}^2}}$$

This approach faces two problems, sparsity and scalability hence the original sparse matrix is decomposed to low dimensional matrices with latent features and less sparsity, results into matrix factorization.

Matrix factorization eventually provides the information about how much a user is aligned with a set of latent features and how much a book fits into this set of latent features. Thus even if two users have not rated any same books, it's still possible to find similarity between them if they have similar taste according to latent features.

Matrix factorization comprises of Singular value Decomposition (SVD).

Any real matrix R can be decomposed into three matrices U, Σ, and V.

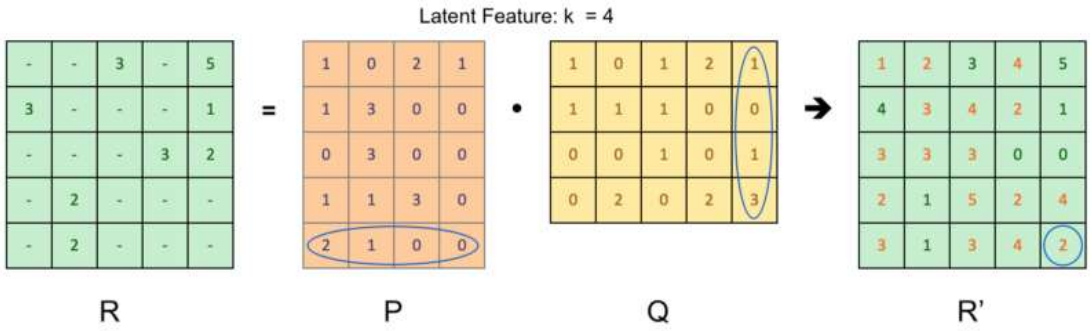
U is an n × r user-latent feature matrix, V is an m × r book-latent feature matrix. Σ is an r × r diagonal matrix containing the singular values of original matrix, which represents how important a specific feature is to predict user preference.

$$R = U\Sigma V^T$$

$$U \in IR^{n\times r}, \quad \Sigma \in IR^{r\times r}, \quad V \in IR^{r\times m}$$

Iteratively we find U and V such that when multiplied back gives an output matrix R' which is closest approximation of R and no more a sparse matrix. Any item i.e. book is denoted as a vector q_i , and user 'u' is denoted as a vector p_u such that dot product of these two vectors is the predicted rating for user 'u' on item 'i'.

$$Predicted\ Ratings: \ r'_{ui} = p_u^T q_i$$



Optimal value of q_i and p_u is defined by a loss function which focuses on minimizing the cost of errors.

$$\min_{q,p} \sum (r_{ui} - p_u^T q_i)^2 + \lambda(\|p_u\|^2 + \|q_i\|^2)$$

r_{ui} is the true ratings from original user-item matrix.

Optimization process is to find the optimal matrix P composed by vector p_u and matrix Q composed by vector q_i in order to minimize the sum square error between predicted ratings r_{ui}' and the true ratings r_{ui}.

Also, L2 regularization has been added to prevent overfitting of user and item vectors.

Collaborative filtering requires least information and gives satisfactory results, but it has few limitations as well.

1. Latent features derived from the data is not interpretable as it can not be compared with any genre. For example , if a person likes horror genre, then other books of horror genre may or may not be recommended to the user.
2. Whenever a new book is published then it is not recommended until it is rated by a substantial amount of users, the model is not able to make any personalized recommendations.

Content-based Filtering :-

It uses similarities in features to make decisions. It compares user interests to product features. The products that have the most overlapping features with user interests are what's recommended here. Content-based filtering makes recommendations by using keywords and attributes assigned to objects in a database.

Pros:

1. The model can capture the specific interests of a user, and can recommend niche items that very few other users are interested in.
2. It doesn't need any data about other users. Recommendations are specific to every user.
3. Recommendations are transparent and very relevant to the user.
4. This type of filtering is easier to create than collaborative filtering.

Cons:

1. This model has limited ability to expand on the users' existing interests.
2. This technique requires a lot of domain knowledge. It is not a very generalized model. For others attributes may be incorrect or inconsistent.
3. Scalability is poor

We have tried to implement this filtering and the results are satisfactory.

We have use some method of information retrieval like TF-IDF (term frequency-inverse document frequency). This term provide weight to the words of document. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. We have created similarity score matrix .User can provide book name and no. of recommendation he want. This model will calculate similarity score between this query and available dataset and books with high score is recommended to users.

Weighted hybrid method :-

1. Content-based filtering does not involve opinions from human to make the prediction, while collaborative filtering does, so collaborative filtering can predict more accurately.
2. However, collaborative filtering cannot give prediction to items which have never been rated by any user. In order to cover the drawbacks of each approach with the advantages of other approach, both approaches can be combined with an approach known as hybrid technique. Hybrid technique used in our model is weighted technique in which the prediction score is combination linear of scores gained by techniques that are combined.
3. It is more generalized model than content and collaborative.
4. We have used this model to find top 10 book for everyone .
5. It can be used to recommend top 10 author and top 10 locality from where most users are.
6. "Weighted average"=(V*R+C*M)/(V+M)
   Where V=total number of rating given to book
   R=Average rating of that book
   C=mean of all rating
   M=total count in 75 percentile

Conclusion :-

Each methods discussed above is performing well and showing relevant results according to the approach. However, every method have some limitations and can be minimised by using a hybrid model which takes care of users' choice along with collective choice of similar users.

Bibliography :-

1. https://towardsdatascience.com/intro-to-recommender-system-collaborative-filtering-64a238194a26
2. https://iopscience.iop.org/article/10.1088/1742-6596/930/1/012050/pdf
3. Introduction to K-means Clustering | by Dileka Madushan | Medium
4. DBSCAN: What is it? When to Use it? How to use it | by Evan Lutins | Medium