Mitigating to user cold-start issue in Recommendations System

Anurag Singh [1], Dr. Subhadra Shaw [2]

{singhanurag.jbp@gmail.com[1], subhadra.shaw@gmail.com[2]}

Department of Computer Sci. & App., AKS University, Satna(M.P.)[1],

Department of Computer Sci. & App., AKS University, Satna(M.P.) [2]

Abstract. This research paper uses a brand new popular model to resolve users' side issues within the recommendation process. the primary cold problem occurs when the target user doesn't have a rating history within the system. Today, the recommendation system has been utilized in various fields. However, they still suffer from various ailments, including cold sores and sparsity problems. The aim of the model is to recommend the highest five items to the user and therefore the performance of the model is evaluated.

Keywords: Recommend System,popularity-based recommendation, cold-start, rmse.

1 Introduction

Recommendation systems aim to capture users‟ interests supported various sorts of clues and help users discover new items [1]. "The recommender system enhances access and leads to recommend appropriate items to users by visible the users raised choice and objective behaviours" [2]. Nowadays, every e-commerce companies try to use the facility of "RS". It is the most significant think about e-commerce and a number of other applications; the RS utilizes machine learning techniques and tools to predict user‟s preference by utilizing their previous shopping information and selecting products among the tremendous amount of accessible items for the users[1] "Cold-start and scalability are common issues in recommendation systems. Cold-start occurs when the system doesn't have a record or rating history of the target user" [2]. The product within the repository keeps on updating. When a brand new product involves the repository, it cannot be recommended as there's no implicit or explicit information this product [3]. "This is often addressed as a cold-start problem within the literature. The cold-start problem may be a new-user cold-start problem where there's a brand new user and there's no information about the user or a new-product cold-start problem where there's a new product and there's no information about the product" [3][4].

THEETAS 2022, April 16-17, Jabalpur, India

DOI 10.4108/eai.16-4-2022.2318161

2 Popularity Based Recommendation System

This type of "RS" may be a system designed to recommend the foremost popular products to the user. It checks for products that are in trend.[6] Generally, the most well-liked products will be found supported by several filters like user ratings, different locations, etc. “Popularity based recommender system doesn't suffer from a cold-start problem because they will even recommend the foremost popular product to each new user based on different filters” [7]. An "RS" supported popularity avoids the “cold-start" problem since it can recommend the foremost popular product to each new user-supported several filters.[6] No need for user history data. they'll recommend the identical products to all new users visiting the website at that point [5].

3 Literature Survey

There are different approaches and techniques were developed by the researchers for effective product recommendations. **John O’Donovan et. al.[6],** they defined trust because the percentage of true predictions made by a profile normally or for a selected item. They've discussed some alternative ways to mix these distinct varieties of trust values into a typical popularity model algorithm, and they've compared them to a tried-and-true benchmark methodology and a typical data set. **Schafer, J. Ben et al.[7]** proposed demographic-based recommendation system. “This includes past buying behaviour together with their personal profile for predicting their future buying behaviour, the advantage of this approach is it will adapt to the various domains and different types of consumers”[7]. But the extraction of previous user buying behaviour may generate privacy issues. Unavailability of the historical data may create further issues. **Jamali Mohsen, et. al.[8]** presented the trust walker, “a random walk model that a trust-based and recognition model recommendation. A collaborative recommender system cannot offer recommendations for that user who merely gave ratings to a few products”[8]. The random navigation model enables us to quantify and define the recommendation's confidence. The Trust walker adds the trust-based and therefore the popularity model method of filtering to the advice. This model is troubling from the cold start issue and identification of the domain. **Anand et al.[9]** proposed “the metric method supported the user action history to alleviate the item-side cold-start problem”. **Wang et al.[10]** deployed information from a web shopping domain. They attempted to make a cross-domain system. supported the shared users within the system, they attempted to resolve cold-start problems. **Heymann P, et al. [11]** in this paper “Algorithms for recommending non-personal tags tend to duplicate the identical tags for some specific items i.e. always popular tags are recommended for all users” [11]. The customized recommendation completely avoids thunderstorms and all told cases, the foremost popular user history is provided. **Hotho A., et al. [12]** a personalised tag system like Adapted PageRank is an example. They transformed the well-known PageRank algorithm into a folksonomy challenge.

4 Proposed Methodology

The following steps are involved in building a “ popularity based recommendation system” (a) Download the specified data set: There are many datasets available on the web for building recommendations like the assistance reviews dataset, movie lens dataset, amazon ratings_Beauty[16]. This paper deals with the recognition recommender system. The dataset consists of 1048575 data set consisting of user-id, product-id, rating, and timestamp. I considered 60000 data set of total data set. (b) Create a "popularity-based model" and earn points: in a very popular-based recommendation system first, calculate what number of users have rated a product. within the next step, the code divides the database into a training database and a test database using an 80-20 scale. "Stored the number of ratings cherish a respective product in an exceedingly very column and gave it index as "score". The score described count of the rating users has rated to the particular product"[17]. Then it had been grouped in descending order with relevancy products and displayed the foremost popular products which had the best ratings which were then recommended to every user[18]. Product predictions are made based on popularity. (c) Recommendation.

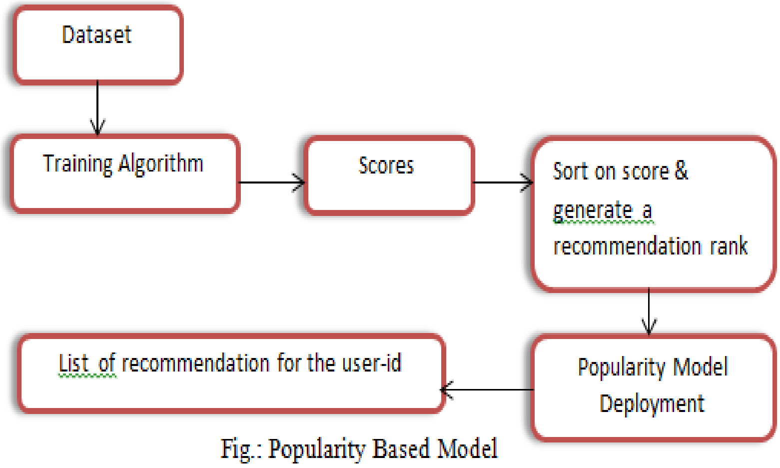


Fig.: Popularity Based Model

4.1 Proposed Algorithms

1 Import the python libraries: Numpy, Pandas, MAtplotlib, sklearn
2 Read the "CSV" information as data frames in user and products -ID.
3 Find the minimum and maximum ratings
4 Check the highest 10 users supported ratings
5 Split the information into the training set and test set as data frame into the variables rating and rating test.
6 Building Popularity Recommder model, Count of user_id for every unique product as recommendation score.
7 Sort the products on recommendation score
8 Use a popularity-based recommender model to create predictions
9 RMSE is calculated to judge the accuracy of the model

5 Experiment Setup and Results

In order to hold out experiment implementation. I've got used a system 4GB RAM, 20GB disk and i3 processor with windows 10 software system[15], the software used are anaconda package manager and python software. Worked and results of this implementation Now, let us have a glance at our Python code output for the popularity based recommendation system.

1 Import the python libraries: ex. Numpy, Pandas, MAtplotlib, sklearn

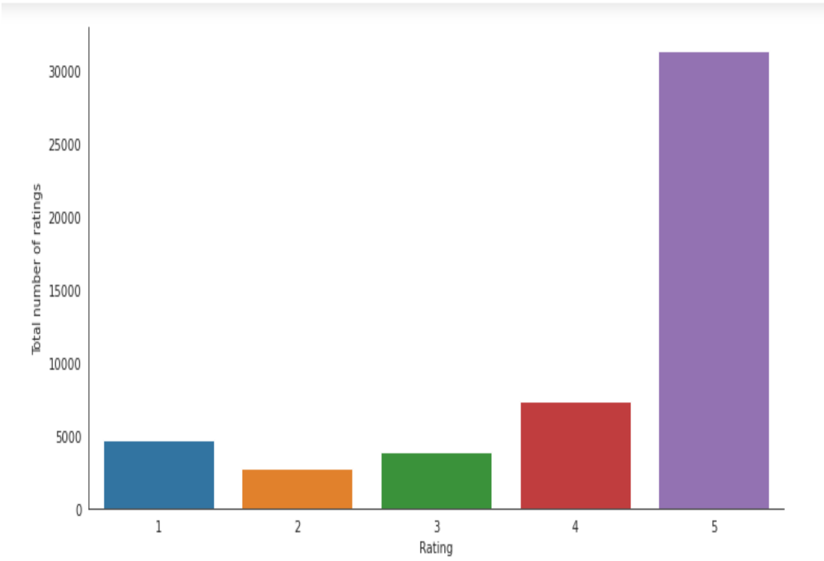
2 Read the CSV information as data frames in user and product -ID. used real-world amazon Dataset: ratings_Beauty.csv

	UserId	ProductId	Rating	Timestamp
0	A39HTATAQ9V7YF	205616461	5	1369699200
1	A3JM6GV9MNOF9X	558925278	3	1355443200
2	A1Z513UWSAAO0F	558925278	5	1404691200
3	A1WMRR494NWEWV	733001998	4	1382572800
4	A3IAAVS479H7M7	737104473	1	1274227200

3 Find the minimum and maximum ratings

Minimum rating is: 1
Maximum rating is: 5

Check the distribution of the rating



4 Check the top 10 users based on ratings

```
UserId
A281NPSIMI1C2R      42
A1Z54EM24Y40LL      25
A3M174IC0VXOS2      22
AK7CAW3N1XZV6       15
AKMEY1BSHSDG7       14
A1BG8QW55XHN6U      13
A2B7BUH8834Y6M      11
A1IU7S4HCK1XK0      10
A22ZFXQE8AWPEP       9
A132ETQPMHQ585       9
dtype: int64
```

5 Split whole data into the training set and test set as data frame into the variables rating and rating test. Split whole data randomly into train and test datasets into 80:20 ratio train_data.

	UserId	ProductId	Rating
26771	A281NPSIMI1C2R	B00006IGL3	5
14223	A1Z54EM24Y40LL	B000053375	5
19726	A1IU7S4HCK1XK0	B00005B703	4
22708	A281NPSIMI1C2R	B00005REB0	5
16420	A281NPSIMI1C2R	B0000536F0	5

6 Building Popularity Recommder model, Count of user_id for each unique product as recommendation score.

	ProductId	score
0	1304139212	1
1	130414643X	1
2	1304174905	1
3	1304511154	1
4	1304622452	1

7 Sort the products on recommendation score

	ProductId	score	rank
78	B0000A4EW3	2	1.0
86	B00012NEYG	2	2.0
0	1304139212	1	3.0
1	130414643X	1	4.0
2	1304174905	1	5.0

8 Use popularity based recommender model to make predictions

Output:

	ProductId	score	rank	UserId
16	B000052XIA	2	1.0	10
34	B00005304H	2	2.0	10
39	B0000530M0	2	3.0	10
40	B000053006	2	4.0	10
43	B000053375	2	5.0	10

	ProductId	score	rank	UserId
16	B000052XIA	2	1.0	20
34	B00005304H	2	2.0	20
39	B0000530M0	2	3.0	20
40	B000053006	2	4.0	20
43	B000053375	2	5.0	20

	ProductId	score	rank	UserId
16	B000052XIA	2	1.0	30
34	B00005304H	2	2.0	30
39	B0000530M0	2	3.0	30
40	B000053006	2	4.0	30
43	B000053375	2	5.0	30

Since it's a Popularity recommender model, so, all three users are given identical recommendations. Here, we predict the products supported their popularity. it's not personalized to a selected user. it's a non-personalized recommender system.

6 Evaluation done on recommendation

This model evaluated based on the metric RMSE." It measures the average magnitude of the error. It is calculated by taking the squared differences between the original values and predicated values" [13]. In this paper we adopt the evaluation measure recommendation system, the RMSE, which is defined as[13]

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(r_{ui} - \bar{r}_{ui})^2}{n}} \qquad (1)$$

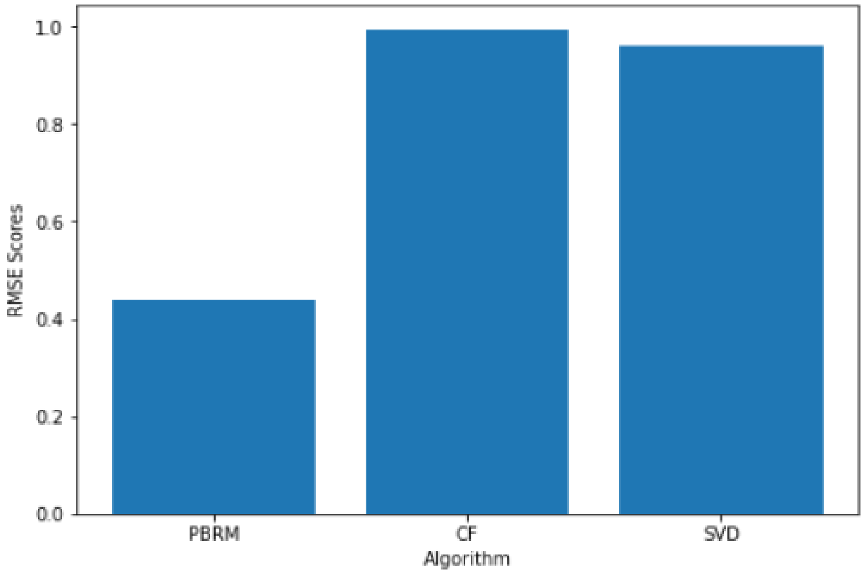
Where 'n' is the predictive value, ' r_{ui} ' is the real product rating[13] 'i' given to the user 'u' and the ' $\overline{r}_{ui}$ ' is the subsequent predicted rating. The lower RMSE value [14] indicated that the "RS" was more efficient than the high value of this. RMSE is calculated to assess model accuracy.

	UserId	ProductId	true_ratings	predicted_ratings
0	A2B7BUH8834Y6M	B00006D2RM	5	5.0
1	A281NPSIMI1C2R	B00012NEYG	5	5.0
2	AKMEY1BSHSDG7	B0000536F0	5	5.0
3	A3M174IC0VXOS2	B0000535U2	4	5.0
4	A1IU7S4HCK1XK0	B00013YYS0	5	5.0

. RMSE value given by the popularity Recommender model was 0.44721.

7 Result and discussion

Firstly tests are performed for PBRM (Popularity Based Recommendation Model) on Amazon ratings_Beauty.csv dataset. RMSE value for PBRM as 0.44721. The RMSE value for collaborative filtering by KNN with means is 0.9951 and SVD is 0.9606. The RSME score of PBRM is better than CF and SVD.



8 Conclusion and Future scope

This recommendation system will help new small-scale industry improve their customer experience on e-commerce and result in better customer acquisition and retention. the advice system, which we've designed relies on a brand new customer's journey from the instant they first visit the business website to their next purchase. Popularity-based could be a vast strategy to focus on the new users with the foremost popular products sold on e-commerce and is extremely useful to cold start a recommendation engine. Build Popularity Recommender model and located the RMSE value for Popularity Recommender model is 0.44721. It's a non-personalized recommender system. Popularity based algorithm have their used cases when the user would similar to browse the foremost popular items. in the near future, the "RMSE" value of "PBRM" could be reduced and also the performance of this model could be improved. We hope this approach will improve the performance of our algorithm because our preliminary experiment has shown promising results.

References

[1] Schafer, J. Ben, Joseph A. Konstan, nad John Riedl. : E-commerce recommendation application,Application of data mining to electronic commerce, Springer US,115-153 (2001)

[2] Guo. G., Zhang, J., and Thalmann, D.: Merging trust in collaborative filtering to alleviate data sparasity and cold start, Knowledge Based Systems, vol.57, pp.57-68, Elsevier (2014)

[3] Cheng, J., & Zhang, L. : Jaccard Coefficient Based Bi-clustering and Fusion Recommender System for solving Data Sparsity , In Pacific-Asia Conference on knowledge Discovery and Datamining ,pp.369-380, April 2019 ,Springer Cham (2019)

[4] Parvin, Hashem, Parham Moradi, and Shahrokh Esmaeili : Nonnegative Matrix factorization Regularized with Trust Relationships for solving Cold-start problem in Recommendation Systems, 26th Iranian Conference on Electrical Engineering, ICEE,2018,may-2019:1571-76(2019)

[5] Y. Zhou, D Wilkinson, R. Schreiber. And R. Pan, " Large-Scale Parallel Collaborative Filtering for the Netflix prize". In Algorithmic Aspects ion Information and Management,2008,pp. 337-348 (2008)

[6] John O'Donovan Barry smyth : Trust in recommender system , Proceedings of 10th international conference onintelligent user interface, January 10-13,2005, san diego, California, USA(2005).

[7] Schafer , J. Ben, Joseph Konstan, and JohnRiedl. : Recommender system in e-commerce, proceddings of 1st ACM conference on Electronic commerce. ACM,(1999).

[8] Jamali Mohsen, Ester Martin : Trust Walker: a random walk model for combining trust based and popularity model recommendation, Proceeding of the 15th ACM SIGKDD, International Conference on Knowledge Discovery and data mining, KDD(2009)

[9] Anand S.S. and N. Griffiths : A market-based approach to address the new item problem", in proc. 5th ACM Conf. Rec. Syst.(RecSys),pp-205-212,(2011)

[10] Wang H., Amagata D., Meekawa T., Hara T.,Niu H., Yonekawa K. and Kurokawa M. : Preliminary Investigation of alleviating user cold-start problem in e-commerce with deep cross-domain, in Proc. World Wide Web Conf., San Francisco, CA,USA, May 2019,pp. 398-403 (2019)

[11] Karatzoglou A.,Amatriain X., Baltrunas L., and Oliver N. : Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering, In proceedings of the fourth ACM conferences on Recommender systems, (2010)

[12] Liu N. N., Cao B., Zhao M., and Yang Q : Adapting neighborhood and matrix factorization models for context aware recommendation. In Proceedings of the Workshop on Context-Aware Movie Recommendation,(2010)

[13] Heymann P., Ramage D., and arcia-Molina H. G : Social tag prediction in Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2008, pp. 531-538 (2008)

[14] Hotho A., Jiischke R., chmitz S C., and tumme S G. : Folkrank: A ranking algorithm for folksonomies, Proc. F GIR, vol. (2006).

[15] Fanca Alexandra, Puscasiu Adela, Gota Dan-Ioan : Recommendation Systems with Machine Learning, 978-1-7281-1951-9/20/$31.00 ©2020 IEEE (2020)

[16] Rodpysh V. K ,Mirabedini J.S.,Banirostam T, "Employing singular value decomposition and similarity criteria for alleviating cold start and sparse data in context-aware recommender systems", © The Author(s), under exclusive license to Springer Science+Business Media, LLC, part of Springer Nature (2021)

[17] Sharma M.,Pant B., Singh V. : Demographic profile building for cold start in recommender system : A social media fusion approach, 2214-7853 / © 2021 Elsevier Ltd (2021).
[18] Ajaegbu C. : An optimized item-based collaborative fltering algorithm, © The Author(s), under exclusive licence to Springer-Verlag GmbH, DE part of Springer Nature (2021).