International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume 5 Issue 4, May-June 2021 Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

Structural Balance Theory-Based Recommendation for Social Service Portal

G. Banupriya, M. Anand

Department of Computer Science and Engineering,
G.K.M College of Engineering and Technology, Chennai, Tamil Nadu, India

ABSTRACT

There is enormous data present in our world. Therefore in order to access the most accurate information is becoming more difficult and complicated. As a result many relevant information gets missed which leads to much duplication of work and effort. Due to the huge search results, the user will generally have difficulty in identifying the relevant ones. To solve this problem, a recommendation system is used. A recommendation system is nothing but a filtering information system, which is used to predict the relevance of retrieved information according to the user's needs for some criteria. Hence, it can provide the user with the results that best fit their needs. The services provided through the web normally provide huge records about any requested item or service. A proper recommendation system is used to separate this information result. A recommendation system can be improved further if supported with a level of trust information. That is, recommendations are prioritized according to their level of trust. Recommending appropriate needs (social service) to the target volunteers will become the key to ensure continuous success of social service. Today, many social service systems does not adopt any recommendation techniques. They provide advertisement or highlights request for a small commission.

KEYWORDS: *Recommendation System, Information System, Social Service, Dataset*

How to cite this paper: G. Banupriya | M. Anand "Structural Balance Theory-Based Recommendation for Social Service Portal" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-5 | Issue-4, June 2021, pp.270-273, URL: www.ijtsrd.com/papers/ijtsrd41216.pdf

IJTSRD41216

I. INTRODUCTION

A recommendation system is an engine which recommends new items to the users by analyzing their preferences. The web contains a huge amount of information like ratings, reviews, feedback on items and other unstructured data. These details are used to get meaningful information of users. There are two common approaches being used to make recommendations (e.g.) collaborative filtering and content-based filtering. In this paper we introduce a hybrid recommendation technique for Big Data Systems. This is an approach in which it combines both collaborative and content-based filtering techniques to recommend items that a user would most likely prefer. It additionally uses items ranking and classification technique for recommending the items.

Currently if you want to get involved in any kind of Social service activity you are required to be part of an organization but the involvement is restricted by the scope and interests of the organization like few focus on children and few on old age etc.. Which will not be able to bring out your full potential or not be able to use properly your time/skillet? Other option to provide social

Service is to donate money on online service which is not for all and only a selected few can help. At times the end user would require things in kind but are forced to request help in way of donation and then they need to procure the necessary items from the market , the problem with this approach is third party websites take their cut which will reduce the amount received by the end user . We are trying to build a website which will involve people to provide services based on their interests and help in ways other than just donating money. To get better of user involvement we plan to provide right recommendations so that the user is able to utilize his full potential and help in any way he can.

II. RELATED WORK

There are many researches for the recommendation system with the help of big data.

Survey 1: "A Survey on Recommendation System for Bigdata using MapReduce Technology" In 2019, Maulik Dhamecha, Krupa Dobaria, Dr. Tejas Patalia, [1]. The proposed work introduces a way to let user interact with the BIG DATA so as to give some recommendations to user based on keywords.

Simple methodology and fast and useful to suggest recommendations based on user keyword history.

Survey 2: "Big Data Analytics for Personalized Recommendation Systems", In 2019, Carson K. Leung, Carson K. Leung Yeyoung Won, and Justin M. C. Choi,

[2]. In this paper they say that, Big data analytics are in demand as they assist users to analyze the big data and discover new knowledge. This in turn helps users gain insight into big data, obtain useful info and make personalized recommendation

Survey 3: "A Hybrid Recommendation Technique for Big Data Systems" In 2018, Chitra Nundlall, Gopal Sohun, Soulakshmee Devi Nagowah

[3]. The paper aims to introduce a hybrid recommendation technique for Big Data Systems. The approach combines collaborative and content-based filtering techniques to recommend items that a user would most likely prefer .

III. SYSTEM ARCHITECTURE

In Social Service Domain the number of categories and user data is limited and thus we don't have enough data to train an ML Model and it is always problematic to fresh users and users who had bad experience with services they provided.

To tackle this we are using a custom model to derive recommendations to users based on their interests and ratings.

The User once he logins and reaches his home page, We load the user information and ask the recommendation System to provide relevant recommendations based on his previous likes and dislikes.

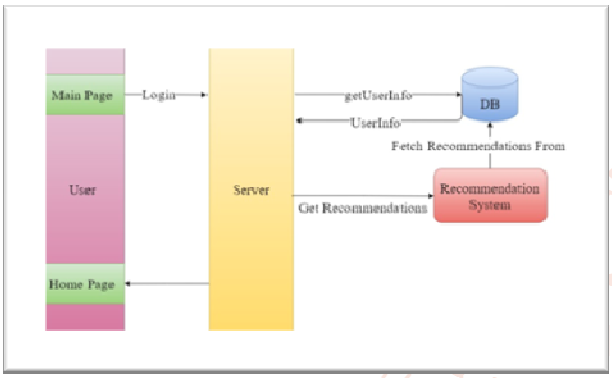

Fig 1: Top Level Design

For the Recommendation System Design, since we need to display the recommendations as soon as the user logins, it needs to get real time recommendations from the system. Thus we decided to train the data for every given interval or on demand. At this time we process the data in the database at that given point, we try to find likes and dislikes of user based on his given rating for a given subcategory of Services he provided and based upon the same we try to get top 3 liked subcategories and bottom 3 subcategories

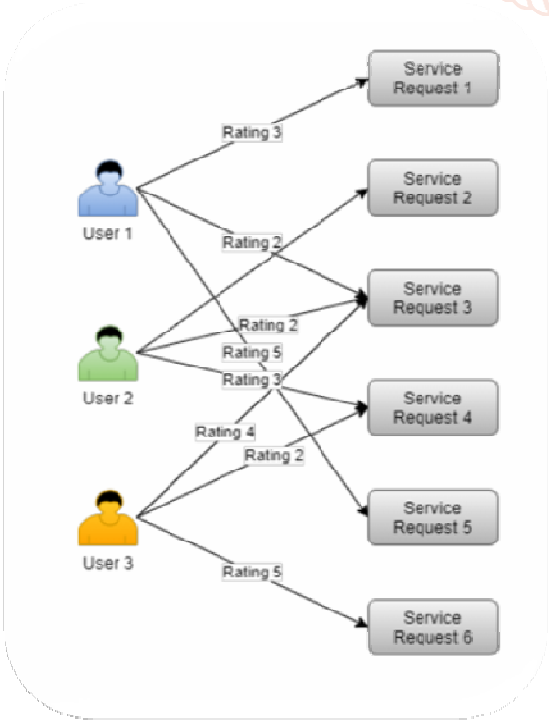

Fig 2: User Rating Service Requests

Top 3 subcategories are selected provided that the average rating of all the rated service responses is greater than a threshold and in the same way we select bottom 3 subcategories provided that the average rating for that subcategories is below a given threshold. If there are no matching criteria then they are taken as null for our calculation purposes.

Once we have the relevant data we try to find a similar enemy of the user based on user bottom 3 subcategories. We are using apache spark with Hadoop to load the data and do the necessary computations and in regards to the algorithm, we could use different approaches but we decided to go with the closest friend by using given formula which is simple and suffices to our use case.

1. If user least favourite subcategory is part of any users bottom 3 subcategories we give 5 points
2. If the user's second least favourite subcategory is part of the user bottom 3 subcategories we give 4 points
3. Lastly if users third least last favourite subcategory is part of the user bottom 3 subcategories we give 2 points

Once we have this information we then sum up the total and then order by descending to find the most similar enemy and then once we have it we retrieve his top 3 favourite subcategories and then add all this info into a different table which would have

- User
- Top 3 subcategories
- Bottom 3 subcategories
- Similar enemy
- Suggested subcategories

This table is used by recommendation system to recommend Service Requests to user.

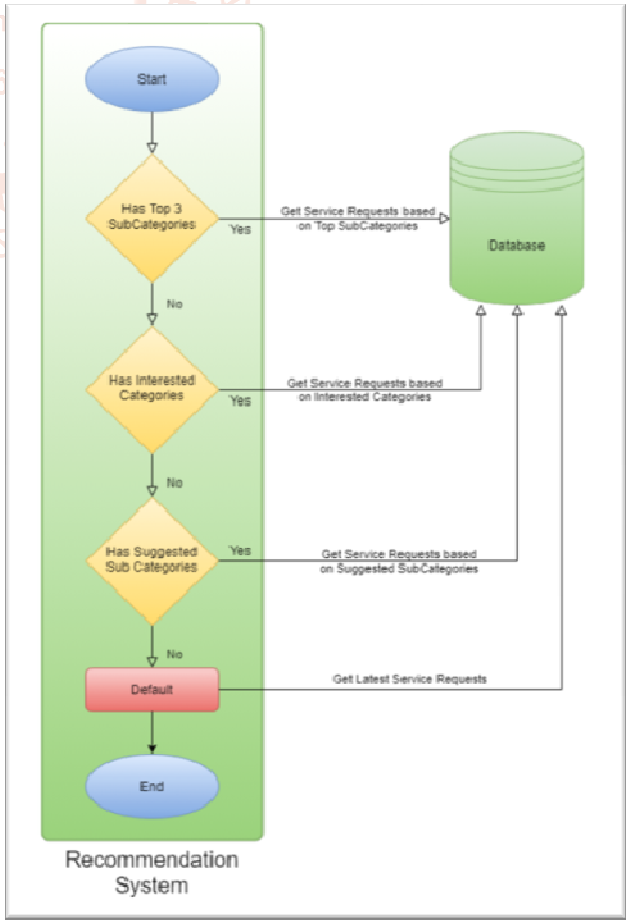

Fig 3: Recommendation System

From Fig 3 we can see the basic outline of the algorithm being used. The typical flow of recommendation system is pretty straight forward.

For a given user we check if he has top 3 subcategories. Even if he doesn't have 3 but has only 1 or 2 top subcategories then we try to fetch recommendations based upon it.

If he doesn't have top 3 subcategories then we check if user has interested categories in his user profile. If present we try to provide recommendation based upon his interested categories.

If he doesn't have interested categories as well then typically we can try to find friend of friend and suggest based upon it but since in this algorithm we are looking at subcategory level and not at Service Request level so we can skip this as if user has top Rated Service Requests they would ensure he has top3 categories.

Now since we don't have friends concept, we are going with enemy of enemy and since we already have a table populated for us with similar enemy and suggested categories based on it. We provide recommendation based upon the suggested sub categories.

There could be cases where we might not even have suggested sub categories and in those cases we fallback to the default recommendation which is to provide some service requests. We could choose either the latest or the top rated Service Requests. In this project we decided to go with latest Service Requests added, that way user would see newer entries every time he logins instead of seeing same set had we gone with top rated ones.

For the UI and Services we are going with simple MVC concept where user would call controller and controller decides what needs to be done and populates appropriate models and redirects to relevant views.

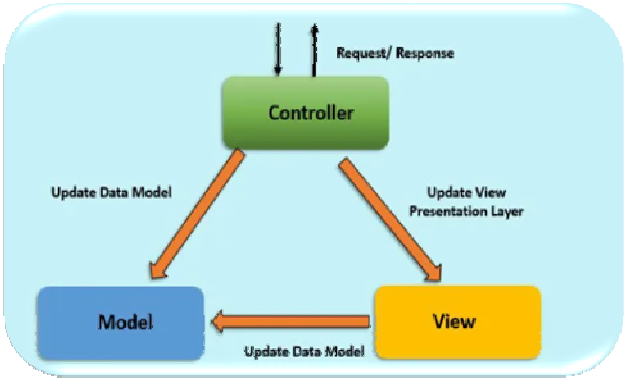

Fig 4: MVC Pattern

In Social Service getting test data has been a problem and thus finding the right dataset has been a bigger problem. There are few datasets present over the internet but they don't have rating system and lot of data is garbled due to their own interpretation of how the data is to be stored in database and how the foreign key constraints are handled. Above all the dataset is very small to provide any meaningful insight or information

To handle the above issue we had to go with Random Generators to populate tables. For users we could generate usernames from online along with few basic details and added interested categories based on Random Integers. We created around 1000 users for our test dataset, Now for Service Requests we added a list of Requests for each Category and Subcategory. For every user we started generating random number of requests for the above list with a upper limit of 100 requests per user and for every request we are populating service Responders randomly with an upper limit of 100 per Request. With this we would have on average of around half a million records of Service Responses along with some having ratings in the range of 0 to 4 and for average calculations we decided not take 0 into consideration and for threshold of average, we decided for a sub category to be considered top sub category its average has to be above 3.0. Once all this done we took this dataset as our test dataset for our project on which we can run our analysis and provide recommendations for all the logged in users.

IV. RESULTS AND DISCUSSIONS

userName	topCategory1	topCategory2	topCategory3	bottomCategory1	bottomCategory2	bottomCategory3	friend	suggestedCategory1	suggestedCategory2	suggestedCategory3
AbhinavaDutta	Food	Maths	Events - Others	Medicines	PhysicalEducation	English	RandySullivan	Events -Others	Cycling	PlantingTrees
AbhinavaKapur	Abacus	NULL	NULL	Marathon	Languages	Clothing	SushilaRao	Provisionals	PhysicalEducation	Volunteers
AdamBarnes	Marathon	Volunteers	Computer	English	Maths	PlantingTrees	LillianIrwin	Computer	Awareness	PhysicalEducation
AdamSykes	Provisionals	Maths	Computer	PhysicalEducation	Abacus	Cycling	KumarNayar	Food	English	Music
AdamWilder	Volunteers	Marathon	NULL	English	Events -Others	Computer	JacobRios	PlantingTrees	Marathon	Drawing
AddieWhite	English	Medicines	PhysicalEducation	Computer	Cycling	Provisionals	Marion	PhysicalEducation	Music	Medicines
AdelineRoberts	Maths	Cycling	Abacus	Clothing	Awareness	Computer	KellyRall	Marathon	Events -Others	Drawing
AdhijaBhatnagar	Food	Events - Others	Maths	Provisionals	In Kind -Others	Computer	ScottHampton	Abacus	Maths	Clothing
AdhijaJayaraman	Provisionals	In Kind - Others	Events - Others	Cycling	Music	Maths	JenniferCooper	Medicines	Abacus	Teaching - Others
AdhijaPawar	Food	Cycling	Volunteers	Clothing	Marathon	Music	Skrikant	Drawing	Abacus	Languages

Fig 5: User Analytics Table populated by the Spark Job using our algorithm

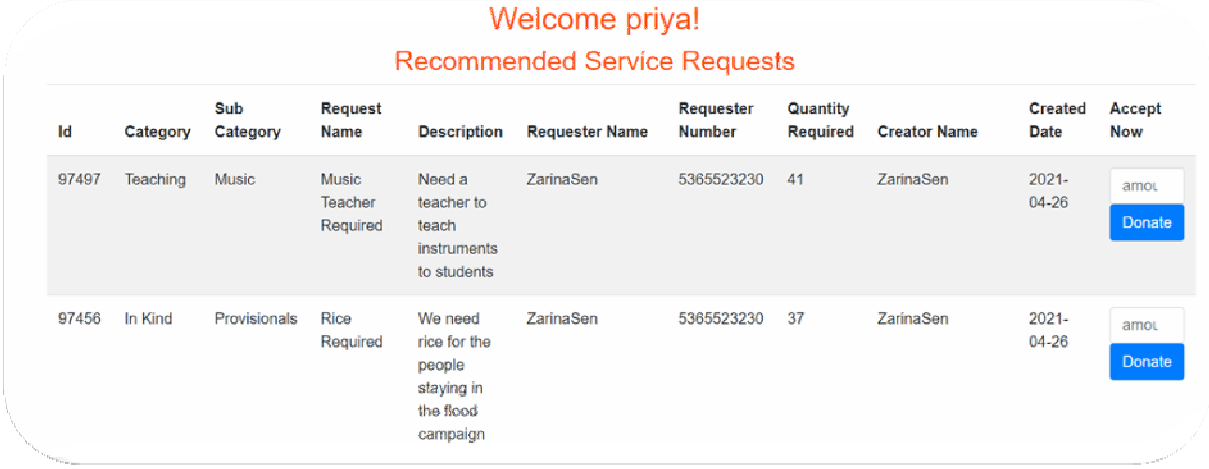

Fig 6: User Home Page with Recommended Service Requests

You can see from Fig 5 we are populating a table called user analytics and we using the same for our recommendations. By using this technique we are ensuring that we are giving relevant recommendations for all the users once they login and if we are unable to give based on the user analytics table we would fallback to give default suggestions by displaying latest requests in the system.

This gives a good user experience to the user and he would be able to start responding to service requests of his choice pretty easily from his home screen only. On an average we are providing decent recommendations to all the users and to test suggested Service Requests we have removed few users top3 sub Categories randomly and added couple of fresh users with only negative feedbacks. We see that the algorithm works fine and gives good results.

That being said there are rare cases, may be less than 1 percent where we found that one of the suggested recommendations could be from user bottom3 Sub Categories since for similar enemies we are considering only one and not more than that. To enhance the algorithm we could take closest 5 similar enemies and then take the top subcategories from the combination of the similar enemies and also ensure none of them are part of users bottom 3 categories are part of the same. This would improve the user experience for even those 1 percent of the user base but the additional computation may not be worth the cause since it is rare scenario and even then he would have recommendations from other 2 categories and thus the user experience would still be good.

V. CONCLUSIONS

The purpose of the project is to provide recommendation to users based on their likes and dislikes and given that there is very sparse data in the social service domain, so we decided to provide recommendation based on subcategories and user interested categories. This might change once we start seeing more data being added when more real users start using the website for few years, then we can use machine learning models to recommend service requests to user based on rating he provided for service requests instead of providing recommendations on subcategory and category level. The downside would be we would need more Hadoop nodes and high end data processing machines to crunch the number and train the model to get decent recommendations.

In a domain where every penny saved is every penny used to social service it could be a challenge for a free website with minimal resources to handle but once the website and user base grow to an extend then may be it might make sense to extend this concept to service request level but then again service requests once completed won't have any usage to recommend to new users and thus we might have to take a call once we reach that level in the future.

VI. REFERENCES

[1] "A Survey on Recommendation System for Bigdata using MapReduce Technology" In 2019, Maulik Dhamecha, Krupa Dobaria, Dr. Tejas Patalia

[2] "Big Data Analytics for Personalized Recommendation Systems", In 2019, Carson K. Leung, Carson K. Leung Yeyoung Won, and Justin M. C. Choi,

[3] "A Hybrid Recommendation Technique for Big Data Systems" In 2018, Chitra Nundlall, Gopal Sohun, Soulakshmee Devi Nagowah,

[4] "Accurate and Diverse Recommendation based on Users' Tendencies toward Temporal Item Popularity" In 2017, Koki Nagatani, Masahiro Sato,

[5] "TRUSTED RECOMENDATION SYSTEM BASED ON LEVEL OF TRUST(TRS_LoT)",In 2017 Qusai Abuein, Ali Shatnawi, Huda Al-Sheyab,

[6] "Review Based Service Recommendation for Big Data" In 2016, Khushboo R. Shrote , Prof. A.V.Deorankar,

[7] "Big Data Analysis: Recommendation System with Hadoop Framework", In 2015, Jai Prakash VermaBankim Pateland Atul Patel,

[8] "Novel Recommendation based on Personal Popularity Tendency," In 2011, Jinoh Oh, Sun Park, Hwanjo Yu, Min Song† and Seung-Taek Park