International Journal of Research Publication and Reviews

JOB RECOMMENDATION SYSTEM VIA SOCIAL MEDIA USING MACHINE LEARNING

Ankita Bhosale[1], Pratiksha More[1], Aradhana Sutar[1], Rajababu Swain[1], Guide: Prof. Gajanan Arsalwad[2]

[1]*Bachelor of Engineering, Department of Information Technology,Trinity College of Engineering & Research (TCOER), Pune*
[2]*Department of Information Technology,Trinity College of Engineering & Research (TCOER), Pune*

ABSTRACT

In India, only 1-2% of the total Software Engineering graduates find a job of their interest. The main reason of this crisis is the lack of passion, skill-set, Resources and the Academics required to come-up with the real-world requirements. A student or a developer may be passionate with his skills but usually academic grades are the basic criteria for the recruiters to hire a developer. Hence, keeping this problem in mind we intend to develop a single platform for both software developers as well as for the hiring institutes to make their job easier and less complex. This application will provide all the modern features of a social media for the developer to share his skill-set and problems to the IT community. This application will also consist various resources such as Build and Collaboration space where the developer would be able to analyze, manage and track his projects and skills. This application will also have another face which will be for recruiters. The recruiters will be recommended various developers according to their provided requirements. The recruiters can also use the Messaging platform to communicate with the fellow developers.

1. INTRODUCTION

Recommender systems have been widely adopted by many online websites to help users overcome the information overload problem and make their purchase decisions. Popular recommendation techniques, such as collaborative filtering (CF) and content-based filtering, can be utilized in social media for users to search jobs. CF makes recommendations based on the assumption that users with similar ratings and interests for some items are likely to have similar preferences for other items. Content-based filtering assumes that the users' interests are able to be represented by the content of posts they have shown interest in, and those contents that have content descriptions similar to the target user's favorite jobs are recommended.

In this paper, we capitalize on user posts to understand the requirements from the users' perspective and leverage user reviews to develop a job recommender system. The topics hidden in the review texts can be a type of representation of the job features. The topic distributions of apps and user preferences are both considered when producing recommendation scores of the relevant jobs for the user to make personalized recommendations.

2. LITERATURE SURVEY

A learning style questionnaire was presented to the students. Based on the results using Felder-Silverman learning style model, individual learning style of the student was predicted by authors S. Graf, Kinshuk, Tzu-Chien Liu, in the paper "Identifying Learning Styles in Learning Management Systems by Using Indications from Students' Behaviour" [2]. In this Rule based mechanism was used with some limitations such as only specific learning styles were considered.

In the paper "Advanced Adaptivity in Learning Management Systems by Considering Learning Styles" by authors S. Graf and Kinshuk [3] adaptive learning mechanism which provides students with courses that best fit their individual learning style was provided where the technique Adaptive generic mechanism is used where constant track has to be maintained as the learning style of students changes.

Paper Title	Publication Year	Publication Journal	Author
Fast Collaborative Filtering with a k-nearest neighbor graph	2019	IEEE	Youngki Park,Sungchan Park,Sang-goo Lee,Woosung Jung
Cryptography in (LSB)	2019	Science Direct	Rahmat Sulaiman; Burham Isnanto; Hengki;

Method Using RC4 Algorithm and AES Algorithm in Digital Image to Improve Message Security.			Chandra Kirana
A Combined Predictor for Item-Based Collaborative Filtering	2020	IEEE	Zhonghuo Wu; Jun Zheng; Su Wang; Hongfeng Feng
Effectiveness comparison of the AES and 3DES cryptography methods on email text messages.	2021	IEEE	Rini Indrayani; Subektiningsih; Pramudita Ferdiansyah; Dhimas Adi Satria

Recommender System:

As discussed previously, RecSys are the system that analyses user preference history

and caters them with different options of services related to the requirement. Recommender systems emerged as an independent research area in the mid-1990s(Ricci et al., 2011). In recent years, the interest in recommender systems has dramatically increased. In the Rec-ommendation algorithm, it classifies into four types: Content-based filtering, Collaborative filtering, Rule-based, and Hybrid approaches (Mobasher, 2007; Al-Otaibi and Ykhlef, 2012). Collaborative Filtering (CF): Collaborative Filtering is a technique is based on the human ratings that are given to an item by a user and find similarity between different users who have given similar ratings to an items(Hu and Pu, 2011). The essential operation used here is the memory-based nearest neighbour approach to group users who have a similar interest. As the volume of data grows gradually, there will be high latency in generating recommendations Mobasher (2007); Herlocker et al. (1999). Collaborative filtering has an advantage over content-based filtering techniques, but due to the nature of the hiring process, a job cannot be rated by the user and will not be possible to create a similarity matrix.

Content-based filtering (CBF): These are the most subjective and descriptive based filtering. Content-based filtering can also be called as attribute-based recommender as it uses the explicitly defined property of an item. It is an approach to an information retrieval or machine learning problem. The assumption made in content-based filtering is that user prefers item with similar properties. Content-based filtering recommends items to the user whose properties are similar to the item which the user has previously shown interest. Mobasher (2007) express that drawback of this filtering technique is their tendency to over-specialize in suggesting the item to a user profile as user profiles are relayed on an attribute of the previous item opted by the user. Nevertheless, in the job domain, the job listed in the job board be available only for few days; due to the nature of the domain, the tendency to over-specialize in recommending the same item would not be any problem in the job domain recommender system. In domains like entertainment, user preference are tends to change depending on various factors, but In Job domain, the user tends to look for the job where he can use his previous skills. New recommendation of jobs can be made when there is a change in user

preference, i.e. if a user thinks to change his/her job domain by updating his new skills and the job domain if he/she wishes. Another scenario of new recommendation is when new jobs are listed in the database; system would identify the properties of the job listed, such asjob domain and skills required for the job and matches with the users with a high similarity score.

Rule-based Filtering (RBF): These filtering techniques depend upon decision rules such as an automatic or manual decision rule that are manipulated to obtain a recommendation for the user profile. Currently, the E-commerce industry uses a rule-based filtering technique to recommend an item based on the demographic region of a user, purchase history, and other attributes that can be used to profile an user. A drawback in rule-based filtering is user feeds the information to the system. These inputs are utilized as a description of a user profile or can be considered as a preference of a user, defined by the user. Thus the data acquired is prone to bias. With the age of the user's profile, recommendation tends to hit the saturation and become staticMobasher (2007).

Hybrid filtering (HF): As the title describe, its incorporation of multiple techniques to improve the performance of recommendation. The previously discussed recommendation technique has its weakness and strengths. In order to get a better recommendation and overcome the challenges posed by earlier techniques, this technique is sought after. All the learning/model-based techniques suffer from cold-start in one or other form. It is a problem related to handling a new user or new item. These and other shortcomings of the CF,CBF, and RBF could be resolved by using hybrid filtering techniques Burke (2007); Jain and Kakkar (2019); Dhameliya and Desai (2019).

The surveys conducted by Burke (2002) and Dhameliya and Desai (2019) have identified different types of hybrid filtering techniques that could be used by integrating CF, CBF, and RBF.

1. Weighted: The similarity score obtained from different recommendation components are coupled numerically to get one better recommendation.

2. Mixed: Recommendations obtained from different recommending techniques are put together and presented as one recommendation.

3. Switching: choosing one among the recommendation components based on the scenar- ios where it suits best.

4. Feature Combination: Attributes derived from diverse knowledge origins are fused and supplied to a recommendation algorithm.

5. Feature Augmentation: One recommendation technique is used to compute a set of attributes of user or item, which is then part of the input to the next recommendation technique. Two or more recommendation techniques are serialized to get on recommendation.

6. Cascade: Recommending systems are given strict priority, with the lower priority ones breaking ties in the scoring of the higher ones. Here one Recsys technique refines recommendation of another. There had been attempts to develop a recommendation system by several researchers. One such implementation was done by Rafter et al. (2000). They had devised a hybrid Recsys CASPER for Job finding search engine. They had implemented an automated collaborative filtering module and personalized case retrieval module in their job recommendation system.

    ACF module utilized user behavior information such as read time and activity on the page during his time on the system to profile the user. Similarity measure such as the Jaccard index and other clustering algorithms was used for similar grouping user against target user. Their other module PCR finds the similarity between the user's query and jobs in the system.

    The module computes similarity with a target user's query and jobs from the job case base using different similarity measures. This system has faced sparsity and scalability problems.

3. SYSTEM DESIGN

- Content-aware collaborative filtering is the integration of content-based recommendation and collaborative filtering.

- Our proposed algorithm targets content-aware collaborative filtering from implicit feedback and successfully address the disadvantages by treating the items not pre-ferred by users as negative while assigning them a lower confidence for negative preference and achieving linear time optimization.

- Accuracy is high..

Pre-Processing:

In this study, dataset acquired for the study has attributes filled string column with symbols and stop words. Especially in the column where skills details are present in both dataset. Data pre-processing is our first process in the second tier of our study's three tier architecture. Below is Fig4.8 that shows data pre-processing process flow.

All the user and Job data is in a single comma separated file. In data pre-processing, our goal is to create a user preference matrix for each column. i.e., we will be creating two dimensional matrix with each rows giving detail of user preferred language skill or database skill. The values of one particular skill column is striped from its default form and transformed to be a columns name against user in each row. We will discuss the data pre-processing of each dataset in detail further sections.

The Dataset

For This JAVA Project, we will be building a recommender system in Java using Apache. Apache is an open source project used to create machine learning algorithms. You can use it to implement machine learning techniques like classification, clustering, and recommendation. The work of the recommender system will be to recommend items to a user.

4. SOFTWARE REQUIREMENTS

- Programming Language – JAVA
- IDE – Eclipse
- Language - Java
- Operating System - Any OS Like A Window, Ubuntu.

Technologies You Will Learn By Working On Job Recommendation System

- JAVA

5. DATA SET

On our job recommendation platform, we extract user interactions as pairs of users and jobs. First, we remove users and jobs that have less than two interactions overall. Then, users are filtered out that have a number of over-all lifetime interactions that exceeds the 99th per- centile of all users. We consider such users as out-liars. As click data of users is noisier than the bookmark data, we do not use clicks for the creation of this dataset.

6. PROBLEM STATEMENT

- To develop a complete Job Recommendation system via social media using machine Learning.

- To develop a multi-application platform for recruitment of the developers.

- To develop a social media platform to build a community of the developers.

- A space to manage, track and showcase the skills and projects of the developer.

- A discussion and collaboration room for all the developers.

- An End-to-End Secured messaging system for both recruiters and developers.

7. EXISTING SYSTEM

Following are some of the existing recommendation system used by the top rated book purchasing websites. The existing system make use of conventional algorithms for recommendations.

- In Content based Recommendation Engine, system generates recommendations from source based on the features associated with products and the user's information. Content-based recommenders treat recommendation as a user-specific classification problem and learn a classifier for the user's likes and dislikes based on product features.

- In Collaborative recommendation engines, suggestions are generated on the basis of ratings given by group of people. It locates peer users with a rating history similar to the current user and generates recommendations for the user.

- In Context based Recommendation system requires the additional data about the context of item consumption like time, mood and behavioural aspects. These data may be used to improve the recommendation compared to what could be performed without this additional source of information.

System Architecture

- Content-aware collaborative filtering is the integration of content-based recommendation and collaborative filtering.

- Our proposed algorithm targets content-aware collaborative filtering from implicit feedback and successfully address the disadvantages by treating the items not pre-ferred by users as negative while assigning them a lower confidence for negative preference and achieving linear time optimization.
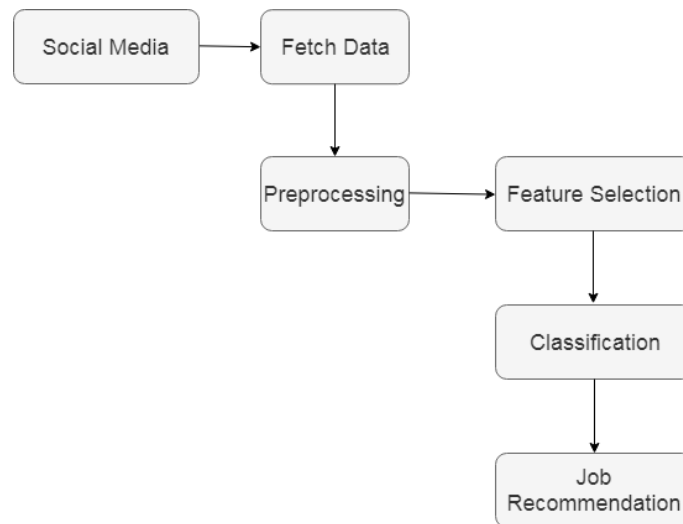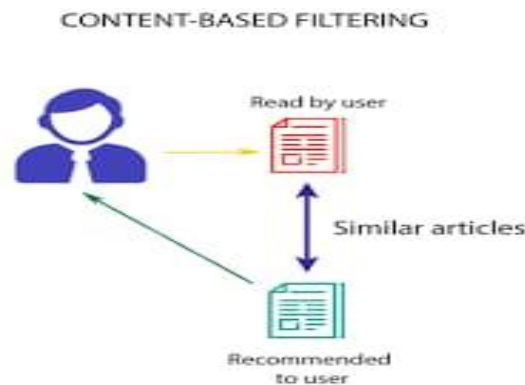
- Accuracy is high.

Figure 6.1: System Architecture
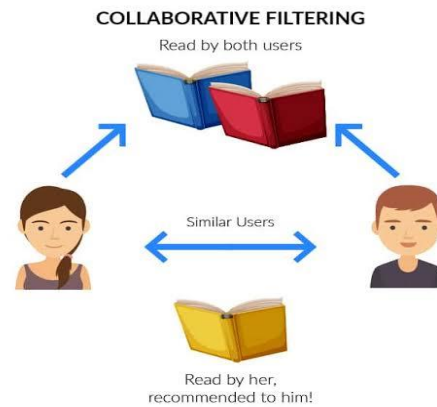
8.   EXISTING SYSTEM ALGORITHM

A.   Content-based Filtering

They are more subjective, and description based i.e. every item to be classified has a different description and suggestions depend upon on the user's profile only. There are keywords/classes that describe and entity to be studied which indicates what type of entity it is and then the type is added to the existing user profile and then suggestions compromise of entities from the identified type and type similar to it.

B.   Collaborative Filtering

Collaborative Filtering, on the other hand defines a group of users into a type on the basis of matching criteria and similarity to other users. One advantage is that because it typecasts users and not objects thus it can recommend a wide range on entities to a group of users after identifying the type of users.

COLLABORATIVE FILTERING

Read by both users

Similar Users

Read by her,
recommended to him!

9.   CONCLUSION

This proposed system gives automated result to reduce unnecessary student or candidate count for the vacant job position. It makes easy, faster way to search the perfect candidate. This system can find the students or candidate information according to his /her percentage, technical knowledge, his communication skill, and experience of industrial work. This application will also consist various resources such as Build and Collaboration space where the developer would be able to analyse, manage and track his projects and skills. This application will also have an another face which will be for recruiters. The recruiters will be recommended various developers according to their provided requirements. The recruiters can also use the Messaging platform to communicate with the fellow developers.

REFERENCES

[1]   [P. Montuschi, F. Lamberti, "A Semantic Recommender System for Adaptive Learning", 2015, IEEE, pp. 50-58.

[2]   S. Graf, Kinshuk, Tzu-Chien Liu, "Identifying Learning Styles in Learning Management Systems by Using Indications from Students' Behaviour", 2008, IEEE, pp. 1-5.

[3]   S. Graf and Kinshuk, "Advanced Adaptivity in Learning Management Systems by Considering Learning Styles", 2009, IEEE, pp. 235-238.

[4]   Y. Li; Z. Wang, "Adaptive reinforcement Q-Learning algorithm for swarm-robot system using pheromone mechanism", 2013, IEEE, pp. 952 – 957.

[5]   K. Chrysafiadi, M. Virvou, "Fuzzy Logic for adaptive instruction in an e-learning environment for computer programming", 2014, IEEE, pp. 1-21.

[6]   Z. Wang , X. Tang, " A Resume Recommendation Model for Online Recruitment", 2015, IEEE, pp. 256-259.

[7]   Y. C. Chen, C. C. Yang, Y. J. Liau, C. H. Chang, "User Behavior Analysis and Commodity Recommendation for Point-Earning Apps", 2016, IEEE, pp. 170-177.

[8]   G. F. Ming, "Course-Scheduling Algorithm of Option-based Hierarchical Reinforcement Learning", 2010, IEEE, pp. 288-291.

[9]   I. Portugal, P. Alencar, D. Cowan, "The Use of Machine Learning Algorithms in Recommender Systems: A Systematic Review", Expert Systems With Applications (2017), pp. 1-30.

[10]   F. Sarro, F. Ferrucci, M. Harman, "Adaptive Multi-objective Evolutionary Algorithms for Overtime Planning in Software Projects", 2017, IEEE, pp. 1-21.