A Machine Learning Approach for Personalized Job Recommendation System

Harini Priya B R, Kannimalar K, Saranya S B, Subbulakshmi B

Thiagarajar College of Engineering, Madurai, Tamil Nadu, India

Corresponding author: Harini Priya B R, Email: harinipriya@student.tce.edu

Every year lakhs and lakhs of students are graduating. The students from top colleges find a suitable job with fewer struggles since most of the companies visits their college for recruitment. But this is not the same case for all colleges. So these students go in search of search engines to find a suitable job. This is not an easy task. Thus a recommendation engine that would recommend jobs based on their search history is built. Currently, more websites give plenty of information about job opportunities that may not intersect. The goal is to ease the work of job searchers by providing them with personalized job recommendations from various websites. This approach tries to overcome the drawbacks of existing papers such as cold start problems, security issues and scalability thus providing a better recommendation system. Thus the recommender system can play a significant role to help college graduates and job seekers to fulfill their dreams by recommending a job based on their interests.

Keywords: Recommendation engine, Machine Learning, Hybrid recommendation.

2021. In Rahul Srivastava & Aditya Kumar Singh Pundir (eds.), *New Frontiers in Communication and Intelligent Systems*, 399–404. Computing & Intelligent Systems, SCRS, India. https://doi.org/10.52458/978-81-95502-00-4-42

1 Introduction

India has the biggest and fastest-growing number of skilled individuals with invents of internet, opportunities are unlimited. In such an environment, the only thing stopping individuals from getting an opportunity of their choice is spending most of their time searching for the right job rather than getting skilled and also in case of working people who want to switch to another job, they don't have ample amount of time to pick a perfect job. There are plenty of websites that provide loads of information about job opportunities. But this task is extremely tedious for fresher's or job seekers as they need to go through many websites to find the perfect job. The recommender systems are more common nowadays. But the type of recommendations provided may be different based on the domain of its use. In the case of the job recommendation system, personalized job recommendations are more suitable. To help job seekers find their perfect workplace, a recommendation system that recommends jobs to users based on their previous searches is designed. There are many job search websites which list jobs posted by recruiters. There are many job search websites like Indeed, Internshala, LinkedIn, etc. One can use these websites and search for jobs and apply through the link provided. These websites demand educational background and personal information. They also have various filter options using which one can minimize the time of search. Even Though it has some good advantages it is not suitable in every situation. But this has some disadvantages such as the recruiters may not post all available job searching websites. So one may miss a great opportunity and it's also a time-consuming process to look into all available websites, giving all the details in various websites is dangerous. Nowadays data leaks happen so commonly. Various papers have been proposed based on the job recommendation system. In paper [1], a recommendation system is provided in which jobs are scraped from websites and then job offers with common attributes are clustered. If a user likes a job that belongs to one cluster, then the jobs that were in the same cluster were recommended and also based on the user's interaction such as liking, applying and so on. The drawback is one can't decide the recommendation based on the user's likes as the interest of the user varies from time to time. In paper [2], they proposed a recommendation system in which job aspirants subscribe to mail alerts for new job postings that will match their job interests. Here recommendation job list is found based on the observations of user click behaviour, with that developed the set of features that reflect the click behavior of individual job aspirants and then it was clustered based on the users. The drawback here is the grey sheep problem which means that there are always some people whose taste doesn't match with anyone. In paper [3], they have provided the analysis of existing different job recommender systems and their techniques and also pointed out the drawbacks in each recommender system such as cold start problem, security issues, and scalability and so on. In paper [4], a hybrid recommendation system is proposed in which data is scraped using a web crawler and applied a collaborative filtering recommendation algorithm and content-based recommendation algorithm. The drawback here is that web scraping has a much more focused approach and purpose while web crawlers will scan and extract all data on a website. In paper [5] the user's skill set is collected in a specific format. Then a data mining approach is followed to match the user's skills with company requirements. Based on job requirements a test is conducted to judge them. This approach gives only a restricted recommendation to the user and thereby demotivating them. Paper [6] approaches a content-based algorithm to rank top candidates using cosine similarity and KNN algorithm.

2 Proposed Approach

The architecture of the proposed system is depicted in Figure 1. It has four layers namely input layer, database layer, recommendation layer and UI layer. The input layer extracts all necessary information like user details and job openings from various websites using a web scraper. The second layer, the database layer stores all the recovered data in a safe and easily recoverable container. With all this information a recommendation engine is created at the recommendation layer which generates top recommendations. The UI layer is a customer point of view that has four screens to list recommendations, to search for suitable jobs, to view the detailed description of a job and to list search history.

The overall workflow of the algorithm is mentioned in Figure 2 and the explanation is mentioned below.

(i) The user's location and preferred domain are collected. Various job details are scrapped from different websites.

(ii) If user history is present for the respective user, jobs that are similar to the history are recommended.

(iii) If there is no user history, popular jobs are recommended in the ascending order of the distance between the location of the job and the user.

(iv) If there is a new job available,

a. If it belongs to a cluster that already has jobs that were recommended to that user, then the new job is added to the recommendation list.

b. If not, it is not added to the recommendation list.

In this approach, there is no security issue since only user history is known about a user. None of the personal information is got from the user. To solve the scalability problem the outdated jobs are removed and user information of inactive users is moved to another database. If the user becomes active again their info can be retrieved from another database. If a new user comes (cold start problem), then top-seen jobs are sorted based on the distance between the location of the job and the user location.

2.1 Implementation

The language used is Python. Python is a interpreted programming language. Python is a language that is often used to build websites and software, automate tasks, and conduct data analysis.

Extracting job details

Job details are extracted from various websites like Amcat, Indeed, Internshala, LinkedIn, etc. Web Scraping is a process of extracting information from websites and online content. It is a free method to extract information and receive datasets. With this dataset further analysis is made. Packages used are Beautiful soup and Requests. Beautiful Soup parses HTML into machine-readable tree format to extract DOM Elements quickly. It allows extraction of a certain paragraph and table elements with a certain HTML ID or Class. Requests would get the HTML element from the URL this will be the input for BS to parse.

Preprocessing extracted job details

Natural Language Processing or NLP is a field of AI that gives machines the ability to read, understand and derive meaning from human languages. All basic preprocessing steps like removing stopwords, lemmatization, converting all texts to lowercase, etc., were done. The package used is NLTK. NLTK, or Natural Language Toolkit, is a Python package that you can use for NLP. Before analyzing the data it must be preprocessed for better results.

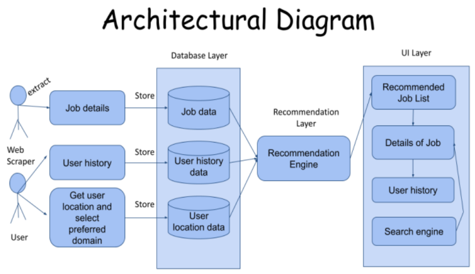

Fig. 1. Architecture Diagram

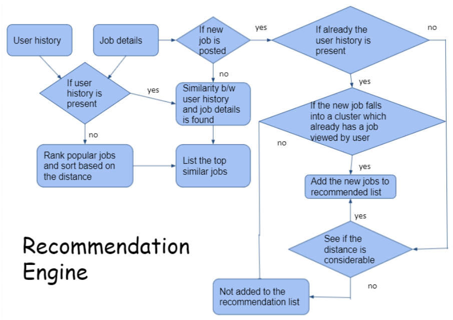
User history
Job details
If user history is present
yes
no
Rank popular jobs and sort based on the distance
If new job is posted
yes
no
Similarity b/w user history and job details is found
List the top similar jobs
If already the user history is present
no
yes
If the new job falls into a cluster which already has a job viewed by user
no
yes
Add the new jobs to recommended list
yes
See if the distance is considerable
no
Not added to the recommendation list
Recommendation Engine

Fig. 2. Workflow of Recommendation System

Recommendation engine

Hybrid content-based recommendation: It finds similarities between job details and user history using the doc2vec model. Doc2vec is an NLP tool for representing documents as a vector.Doc2vec is a very nice technique. It's easy to use, gives good results, and as you can understand from its name, is heavily based on word2vec. The package used is Gensim.

Gensim is a python library for topic modeling, document indexing and similarity retrieval with large corpora. It mainly focuses on the natural language processing (NLP) and information retrieval (IR) community. The significant features of Gensim include that all algorithms are memory independent concerning the corpus size i.e., can process input larger than RAM, streamed, out-of-core, easy to plug in its own input corpus/datastream and easy to extend with other Vector Space algorithms ie., Intuitive interfaces and distributed computing.

Clustering: It clusters similar jobs using the DBSCAN algorithm. DBSCAN is a well-known data clustering algorithm that is commonly used in machine learning. DBSCAN groups together data that are close to each other based on usually Euclidean distance and a minimum number of points. It also marks as outliers the points that are in low-density regions which is a great advantage. The package used is Sklearn. Scikit-learn is a machine learning library for the Python. It features various classification, regression and clustering algorithms.

3 Results

Displayed various job details were scraped from different websites and a user history dataset was created. A dataset with test user history is created. The dataset has features like Jobid, jobTitle and job description. The user history has features like Userid and Jobid. The test dataset has features such as Userid and Jobid.

For testing purposes, the user history of a random user was used. The user has viewed jobs related to software development. The recommendation generated by the algorithm was pretty good. Precision and recall at cutoff k is calculated. The k chosen was 5. Precision and Recall at cutoff k, P@k, and r@k, are simply the precision and recall calculated by considering only the subset of the recommendations from rank 1 to k. Figure 3 is a graphical representation of Precision@k vs Recall@k of the selected random user. From this graph, it can be inferred that considering the top 5 recommendations gives good precision and recall score. The metric used here to determine the performance of the recommendation system is the Mean Average Precision. It is the mean of average precision of all the users. Average Precision rewards you for giving correct recommendations. If there is a need to recommend N items and there are m relevant items in the full space of items, Average Precision AP@N is defined as(1). AP applies to a single user. Mean Average Precision MAP@N just goes a step further and averages the AP across all users. MAP@N is defined as (2). The Mean Average Precision for the recommender system is nearly 88%.

$$\square\square@\square = (1/\square)\sum_{\square=1}^{\square} (\square\square\square\square\square\square\square\square\square@\square\ \square\square\ \square^{\square\square}\ \square\square\square\square\ \square\square\ \square\square\square\square\square\square\square\square) \qquad (1)$$

$$\square\square\square@\square = (1/|\square|)\sum_{\square=1}^{|\square|} (\square\square@\square)_{\square} \qquad (1)$$

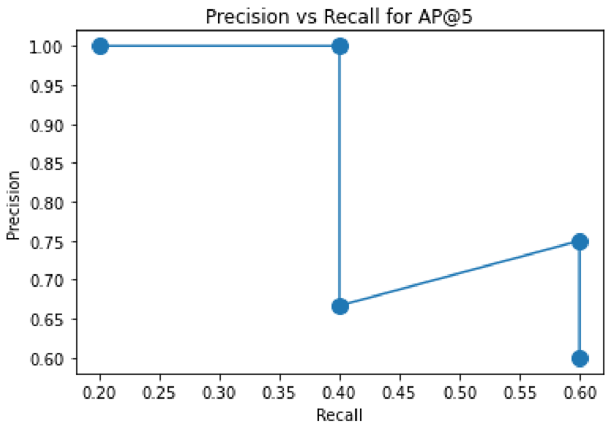

Fig. 3. Sample user's Precision vs Recall graph

4 Discussion

Nowadays it's difficult for fresher's and job seekers to search for a perfect job on many random websites. This type of searching leads to a wastage of time and is difficult to proceed with one's career. To overcome such difficulties, personalized recommendations are given. To achieve this intention, information about the jobs is extracted from various websites and these jobs are recommended based on the job location, user location, user history and their respective job description. Thus providing personalized job recommendations. In the future, this project can be expanded by tie-up with companies to post their jobs in an application and extending the application by recommending courses for the user's career path development.

References

[1] Mhamdi, D. et al. (2020). Job Recommendation based on Job Profile Clustering and Job Seeker Behavior. *Procedia Computer* Science, 175: 695–699.

[2] Jiang, M. et al. (2019). User clicks prediction for personalized job recommendation. *World Wide Web*, 22: 325–345.

[3] Mishra, R. and Rathi, S. (2020). Efficient and Scalable Job Recommender System Using Collaborative Filtering. In *ICDSMLA*, 842–856.

[4] Dong, Z. et al. (2020). *Employment Service System Based on Hybrid Recommendation Algorithm*. Springer, 1:368–375.

[5] Tayade, T. et al. (2020). Data Mining Approach to Job Recommendation Systems. In *International Conference on Mobile Computing and Sustainable Informatics*, 1: 503–509.

[6] Roy, P. K., Chowdhary, S. S. and Bhatia, R. (2020). A Machine Learning approach for automation of Resume Recommendation systems. *Procedia Computer* Science, 167: 2318–2327.