International Journal of Enhanced Research in Science, Technology & Engineering
ISSN: 2319-7463, Vol. 13 Issue 4, April-2024, Impact Factor: 8.375

Bridging The Gap between Job Seekers and Recruiters: The Role of Advanced Job Recommendation System

Subhi Pandey[1], Mohsin Bagwan[2], Arham Shaikh[3], Kutbuddin Khan[4], Amita Raman[5]

[1,2,3,4]Student, Computer Science And Engineering, Saraswati College Of Engineering, Mumbai, Maharashtra, India
[5]Professor, Computer Science And Engineering, Saraswati College Of Engineering, Mumbai Maharashtra, India

ABSTRACT

Machine learning is a branch of data science that focuses on creating algorithms that can learn from and make predictions based on data. These days, recommendation systems are being used to tackle the problem of information overload in various domains, helping users focus on information relevant to their interests. One area where such recommender systems can be particularly useful is in helping college graduates find jobs that match their skill sets. There are many websites that provide a wealth of information about job opportunities, but sifting through all this information to find the ideal job can be a daunting task for students. Moreover, many students are unsure about which jobs are suitable for them. In the current scenario, the IT sector is booming. Many engineering students are acquiring technical skills through various courses, but they often don't know which skills are required for which jobs. At the same time, existing job recommendation systems tend to focus only on the user's domain of interest, ignoring their profile and skill set, which could be used to recommend jobs that are a perfect fit for the user.

Keywords:Cosine Similarity, Job Recommendation, K-Nearest Neighbour, Skill Set, TF-ID Fvectorizer.

INTRODUCTION

According to a recent study, many college graduates struggle to choose the right job domain, especially those engineers who are looking to transition into the IT sector. They often take online courses and search for jobs randomly, without a clear understanding of which IT domain suits them best. To address this issue, the report proposes a job recommendation system that analyzes the skills of candidates and recommends suitable jobs. The system works by reading a candidate's resume and their skills. The resumes undergo a preprocessing stage to enhance the system's efficiency. This involves using a Porter Stemmer to reduce words to their root form and removing stop words, which are words that do not contain important meaning. The system uses a method called TF-IDF (Term Frequency- Inverse Document Frequency) vectorization on both the resume and job descriptions. It then compares the skills listed in the resume with those in the job description. The comparison is done using a function called Cosine Similarity, which calculates the similarity between two documents. The system then scores the resume for each job and sorts the jobs in descending order based on these scores. This gives the candidate a ranked list of jobs that best match their skills. The candidate can then choose a job from the top of the list that aligns with their existing skills. But the system doesn't stop there. It also recommends skills that the candidate could improve to increase their chances of securing a job. This allows the candidate to train themselves for future opportunities and become more accomplished in their chosen domain. The proposed system uses machine learning models to accurately predict suitable jobs for candidates by finding similarities between job descriptions and resumes. This application can be used by any candidate who wants to find suitable jobs and improve their soft and hard skills. It saves them time by eliminating the need to search for jobs randomly. It also helps them grow their skills in their chosen domain, enabling them to progress faster in their career.

METHODOLOGY

This application aims to help job seekers find suitable jobs by analyzing their resumes and comparing them with a dataset of job descriptions. The application first pre-processes the resume and job descriptions by removing stop words and reducing words to their root form using Porter's Stemmer. This results in a meaningful bag of words. The application then uses a method called TF-IDF vectorization to convert the raw text into a matrix, which simplifies the comparison process. The key step is comparing the bag of words from the resume with those from the job descriptions. This is done using the Cosine Similarity function, which calculates the similarity between two documents based on the cosine of the angle between them. The application then ranks the jobs in descending order based on their similarity scores. But it doesn't stop there. The system also identifies skills that the candidate could improve by comparing the resume with a skills dataset and highlighting the skills that are not present in the resume. One of the main contributions of this work is that it makes job recommendation systems, which are currently used by large MNCs and often require a subscription fee, accessible to the average person. This approach could reduce unemployment by helping job seekers find jobs that match their skills, and could also lead to faster growth for companies and more job openings. The goal of the proposed work is to recommend not just one, but a list of jobs that are ideal for the user, and to suggest skills that the user could improve to increase their chances of securing these jobs. This project is intended for anyone who is unsure about their career path.

LITERATURE REVIEW

Existing works are mainly found for the company to select a candidate who is fit for their vacancy [1] There are many experiments for calculating the for recommendation algorithm but with a different distance formula namely the Minkowski distance [2] And some others are tried a different recommender system like collaborative which only helps when there are more data to relate. That won't help for a person who is searching that which job is the correct choice for him/her. R.J. Mooney and L. Roy used Content-Based Book Recommending [3] where the content-based recommendation helps for a cold start. And some authors also say that a content-based recommender is best when they researched a comparison study of job recommendations [4] A recommender system is not only the main part of accurate prediction. There are some other things like vectorizing the words and then similarity functions. Authors like Shouning Qu [5] said that for text mining, tf-idf is the best approach for text feature selection. RavaliBoorugu has researched NLP and tried various text summarization techniques [6] Some papers also say about similarity detection with many languages [7] Jeevamol Joy and Renumol V G discussed which similarity is the best one for a content-based recommending system. They finally concluded that cosine similarity is the best similarity for content based recommended system. Cosine similarity is not only used for recommender systems but is used to find the similarity functions between two sentences or two paragraphs [8]. Tanya V. Yadalam, Vaishnavi, M. Gowda, and Vanditha researched those career recommendations content-based filtering which was mostly like my project but inside it, they mostly discussed security, transparency for the data, and the framework [9].

In many papers, they have been solved through content recommender which is not enough. A literature paper had done research on content recommender system, tf-idfvectorizer, and cosine similarity in a row but in that the author doesn't think about the implementation process and only concentrated more on securing the data. The KNN technique is used in this system, where k is the number of neighbors considered, is a non-parametric method and it infers both nominal attributes the most common attribute value between the k nearest neighbors and numerical attributes the average of the values of the k nearest neighbors [10].

DATASET

The Job Dataset was created by combining resources from Kaggle and Google searches. Additionally, resume data was collected from friends. This dataset contains 958 job descriptions in a .csv format. Each row in the dataset represents a unique job and includes features such as Job Title, Rating, Company, and Description. In addition to the Job Dataset, there is another dataset called the Skill Dataset. The information for this dataset was gathered from Google by searching for each job. This dataset includes the job title and the skills required for that job. It consists of 14 rows, each listing specific skills for a particular job. This way, the Skill Dataset provides a detailed breakdown of the skills needed for different jobs.

SYSTEM DESIGN

A. Overall High Level Architecture

The high-level architecture diagram (Figure 1) shows the overall structure of the system. The system uses a file containing job names and descriptions, and information entered by the user about their skills and experience. These data are read from their respective sources and saved for later use. The system then uses a method called tf-idfvectorizer to convert the raw text data from the job dataset and the resume into a matrix. This matrix represents the importance of each word in the dataset. Next, the system uses a method known as Cosine Similarity to calculate the similarity scores between the job descriptions and the resume. This method calculates the cosine of the angle between two vectors, which in this case are the tf-idf vectors of the job description and the resume. The system first displays the top matching jobs and their similarity scores in a table format. It then visualizes these results in a pie chart for easier understanding. Finally, the system analyzes the user's resume and a skill dataset to suggest skills that the user could improve on to increase their chances of matching with more or better jobs. This system essentially helps users optimize their resumes

International Journal of Enhanced Research in Science, Technology & Engineering
ISSN: 2319-7463, Vol. 13 Issue 4, April-2024, Impact Factor: 8.375

for specific jobs by suggesting improvements based on job descriptions and required skills. It's a useful tool for job seekers looking to tailor their resumes to specific job postings.

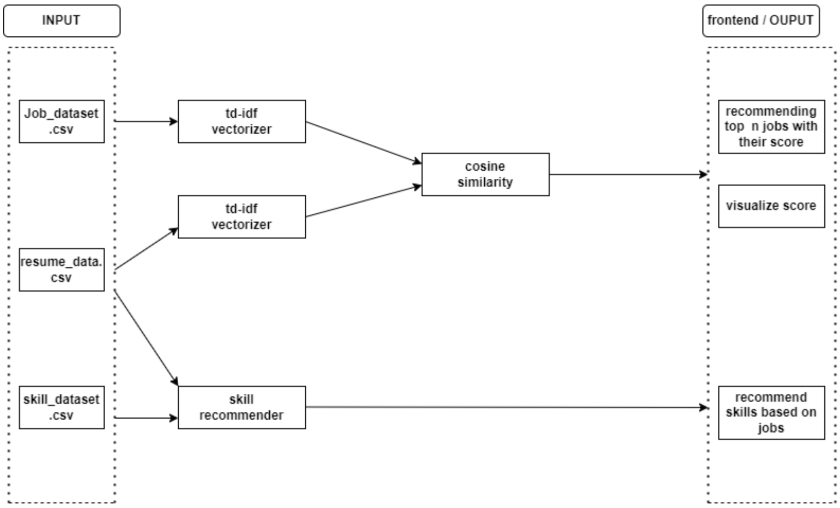

Fig 1: High Level Architecture

B. Preprocessing

The Pre-processing diagram (Figure 2) shows how the system processes the data to obtain a cleaned dataset. This involves using techniques like Porter Stemming and Stop words removal. The job dataset file includes "ID," "Query," "Job Title," and "Job Description". In this context, "Query" is a superset of "Job Title", meaning it contains all the information in the job title and more. Therefore, the system only needs the "Query" and its description. These are common words like "is" or "are" that do not contribute to the meaning or score of the job description. The system removes these words to focus on the important terms in the job description. This is an algorithm used for stemming, which is the process of reducing a word to its base form or root (lemma). For example, "run", "runs", "running", and "ran" are all variations of the same lemma: "run". Porter Stemmer is used to lower the frequency of words in the dataset by converting them to their lemma forms. After the above preprocessing steps, the original dataset is transformed into a new, cleaned dataset that contains only lemma terms. This is a library in Python that provides tools for working with human language data (text). It is used in this system for the data preprocessing tasks.

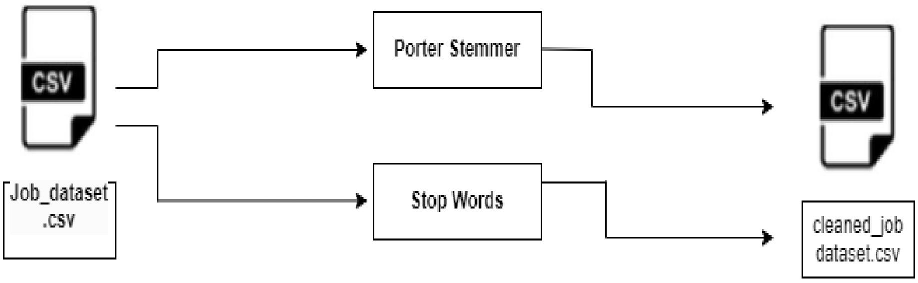

Fig 2: Preprocessing

C. Similarity Function Module

The system obtains a resume from a user in pdf format. There's a validation step that only allows pdf files (.pdf) to be selected by the user. The system uses a cleaned employment dataset, which has been processed to remove unnecessary words and reduce words to their base forms (lemmas). The system uses a method called tf-idfvectorizer to convert the raw text data from the resume and the jobdescriptions into a matrix. This matrix represents the importance of each word in the dataset. The system then uses a method known as Cosine Similarity to calculate the similarity scores between the job descriptions and the resume. The formula for cosine similarity is the dot product of the two vectors divided by the product of their lengths. Once the algorithm is operational, the system can produce scores indicating how closely the job description and the resume match. These scores are then sorted in descending order so that the user can quickly select the top-rated jobs and focus on them. Cosine similarity can be described through equation like this:

$$\text{cosine\_similarity}(A, B) = \frac{A \cdot B}{|| A || \times || B ||}$$

D. K-Nearest Function Module

The K-Nearest Neighbors (K-NN) algorithm is versatile because it can be used for both classification and regression tasks. It's simple because it doesn't require complex mathematical calculations or assumptions about the underlying data distribution. K-NN is a non-parametric method, meaning it doesn't make any assumptions about the underlying data distribution. This makes it a flexible choice for various types of datasets. K-NN can handle both numerical and categorical data. This is beneficial as many real-world datasets often contain a mix of both types of data. K-NN is less sensitive to outliers compared to other algorithms. This is because it makes predictions based on the local structure of the data rather than the overall data distribution. K-NN algorithm works by finding the K nearest neighbors to a given data point based on a distance metric, such as Euclidean distance. The class or value of the data point is then determined by the majority vote or average of the K neighbors. Euclidean distance is a common distance metric used in K-NN. It's the straight-line distance between two points in a plane or hyperplane. This metric helps calculate the net displacement between two states of an object, making it useful for identifying the nearest points or groups for a query point. The formula for Euclidean distance between two points p and q in an n-dimensional space is:

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \ldots + (p_n - q_n)^2}$$

RESULTS

This system works by user uploading their CV on the platform through which system will process and recommends job based on user profile.

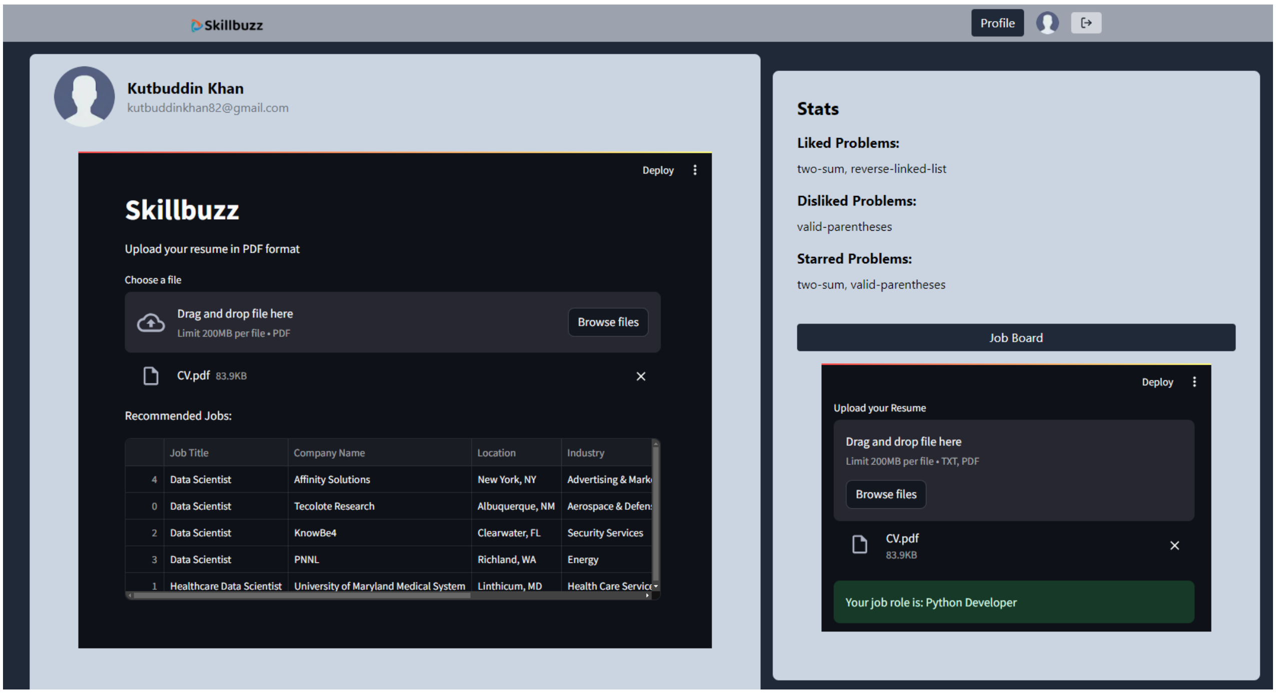

Fig 3: UI Of Job Recommendation System

CONCLUSION

To conclude our team analyzed various research papers and algorithms related to recommendation systems. They have made improvements and modifications to the existing recommendation systems in their proposed system. The job recommendation system takes into account various parameters such as marks, experience, skills, etc. The framework facilitates the understanding of the job recommendation process and allows the use of a variety of recommendation methods according to the preferences of the job recommender system designer. The team plans to perform a more exhaustive evaluation considering a greater number of methods and data. They also plan to comprehensively evaluate the impact of each professional skill of a job seeker on the received job recommendation. Machine learning has been improving the recommendation systems and it brings more possibilities to enhance the performance of the recommendation system. In summary, the team has developed a job recommendation system that considers various parameters and allows for customization. They plan to further improve the system by considering more methods and data, and by evaluating the impact of each skill on job recommendations. Machine learning plays a key role in these improvements.

ACKNOWLEDGMENT

A project is something that could not have been materialized without co-operation of many people. This project shall be incomplete if we do not convey my heartfelt gratitude to those people from whom I have got considerable support and encouragement. It is a matter of great pleasure for us to have a respected Prof.Amita Raman as our project guide, We are thankful to her for being a constant source of inspiration. We would also like to give our sincere thanks to Prof. Shraddha Subhedar, H.O.D, Computer Science & Engineering in Artificial Intelligence and Machine Learning Department, Prof. Dinesh Singh Dhakad, Project Co-Ordinator for their kind support. We would like to express our deepest gratitude to Dr. Manjusha Deshmukh, our principal of Saraswati College of Engineering, Kharghar, Navi Mumbai.

REFERENCES

[1]. M. Alobed, A. M. M. Altrad, and Z. B. A. Bakar, "A Comparative Analysis of Euclidean, Jaccard and Cosine Similarity Measure and Arabic Wordnet for Automated Arabic Essay Scoring," in Proceedings - CAMP 2021: 2021 5th International Conference on Information Retrieval and Knowledge Management: Digital Technology for IR 4.0 and Beyond, Institute of Electrical and Electronics Engineers Inc., Jun. 2021, pp. 70–74. doi: 10.1109/CAMP51653.2021.9498119

[2]. N. D. Almalis, G. A. Tsihrintzis, N. Karagiannis, and A. D. Strati, "FoDRA-A New Content-Based Job Recommendation Algorithm for Job Seeking and Recruiting."

[3]. R. J. Mooney, "Content-Based Book Recommending Using Learning for Text Categorization," 2000

[4]. Sri Venkateshwara College of Engineering. Department of Electronics and Communication Engineering, Institute of Electrical and Electronics Engineers. Bangalore Section, IEEE Computer Society, and Institute of Electrical and Electronics Engineers, RTEICT-2017 : 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology : proceedings : 19-20 May 2017

[5]. Q. Shouning, W. Sujuan, and Z. Yan, "Improvement of text feature selection method based on TFIDF," in Proceedings - 2008 International Seminar on Future Information Technology and Management Engineering, FITME 2008, 2008, pp. 79–81. doi: 10.1109/FITME.2008.25.

[6]. IEEE Staff, 2017 International Conference on Information and Communications (ICIC). IEEE, 2017.

[7]. B. Xu, Institute of Electrical and Electronics Engineers. Beijing Section, and Institute of Electrical and Electronics Engineers, Proceedings of 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC 2016) : October 3-5, 2016, Xi'an China.

[8]. J. Joy and V. G. Renumol, "Comparison of generic similarity measures in E-learning content recommender system in cold-start condition," in 2020 IEEE Bombay Section Signature Conference, IBSSC 2020, Institute of Electrical and Electronics Engineers Inc., Dec. 2020, pp. 175–179. doi: 10.1109/IBSSC51096.2020.9332162.

[9]. Institute of Electrical and Electronics Engineers and PPG Institute of Technology, Proceedings of the 5th International Conference on Communication and Electronics Systems (ICCES 2020) : 10-12, June 2020.

[10]. J. M. Cadenas, M. Carmen Garrido, R. Martínez-España, and A. Muñoz, "A more realistic k-nearest neighbors method and its possible applications to everyday problems," in Proceedings - 2017 13th International Conference on Intelligent Environments, IE 2017, Institute of Electrical and Electronics Engineers Inc., Nov. 2017, pp. 52–59. doi: 10.1109/IE.2017.26.