**Mini Project Report on**



**JOB RECOMMENDATION SYSTEM**



**Submitted in partial fulfillment of the requirement for the award of the degree of**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE & ENGINEERING**

**Submitted by:**

**Student Name Vasu Arora** **University Roll No. 2021524**

***Under the Mentorship of***

**Dr. Narayan Chaturvedi**



**Department of Computer Science and Engineering**

**Graphic Era (Deemed to be University)**

**Dehradun, Uttarakhand**

**January-2025**

GEU logo

**CANDIDATE’S DECLARATION**

I hereby certify that the work which is being presented in the project report entitled **“Job Recommendation System”** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineeringof the Graphic Era (Deemed to be University), Dehradun shall be carried out by the under the mentorship of **Dr. Narayan Chaturvedi,** Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun.

Name Vasu Arora University Roll no 2021524

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Description** | **Page No.** |
| Chapter 1 | Introduction | **1-2** |
| Chapter 2 | Literature Survey | **3-4** |
| Chapter 3 | Methodology | **5-7** |
| Chapter 4 | Result and Discussion | **8-10** |
| Chapter 5 | Conclusion and Future Work | **10-12** |
|  | References | **12** |

**Chapter 1**

**Introduction**

* 1. **Introduction**

The process of job hunting can be both time-consuming and overwhelming, especially with the vast amount of recruiting information available on the internet. Job seekers often spend hours sifting through countless listings, struggling to find positions that match their skills, qualifications, and preferences. This challenge is exacerbated by the fact that job listings are continuously updated, making it difficult for individuals to keep track of the most relevant opportunities. As a result, there is a growing need for more efficient tools to help job seekers find suitable job openings quickly and accurately. In this context, we address the problem of recommending suitable jobs to people who are seeking new employment. The recommendation system proposed in this project is designed to minimize the time and effort job seekers need to spend searching for relevant job opportunities. This approach formulates the job recommendation problem as a supervised machine learning task, leveraging historical data on job transitions, as well as relevant information associated with employees and institutions, to predict the next job transition for a given individual. These systems, which use advanced algorithms to suggest items (such as products, movies, or jobs) based on user preferences or content similarities, have revolutionized many industries, including e-commerce, entertainment, and recruitment. In the context of job search, recommendation systems are able to help users navigate the vast pool of job listings by providing personalized suggestions tailored to their needs.

The main focus of this project is to build a hybrid job recommendation system that combines the strengths of two widely used recommendation techniques: **content-based filtering** and **collaborative filtering**. Content-based filtering works by analyzing the attributes of items—in this case, job listings—and recommending jobs that are similar to those a user has shown interest in. Collaborative filtering, on the other hand, relies on the behaviors of similar users to suggest items that have been well-received by others. By combining these two methods, the hybrid recommendation system can generate more accurate and diverse job recommendations.

Content-based filtering uses the information present in job listings, such as job titles, descriptions, and company names, to calculate the similarity between jobs. A common technique to measure similarity is **cosine similarity**, which calculates the angle between two vectors in a high-dimensional space, where each vector represents a job listing. The cosine similarity score provides a measure of how similar two job listings are based on their content. This approach is particularly useful when users are looking for job opportunities that are closely aligned with specific keywords or job titles they are interested in. For example, if a job seeker previously showed interest in software engineering roles at technology companies, the system would recommend other similar roles based on matching keywords in job titles or descriptions. This technique uses algorithms like **TF-IDF (Term Frequency-Inverse Document Frequency)** or **word embeddings** to represent text and calculate the similarity between job listings. A common method for measuring similarity between jobs is **cosine similarity,** which calculates the cosine of the angle between two vectors in a high-dimensional space representing the job titles and descriptions. If two jobs have a high cosine similarity score, they are deemed similar and can be recommended together.

Collaborative filtering, in contrast, focuses on the interactions and preferences of users. Collaborative filtering can be **user-based** or **item-based**, with user-based collaborative filtering recommending jobs based on the preferences of similar users and item-based collaborative filtering recommending jobs based on the similarities between job listings that users have interacted with. This method can uncover hidden patterns and preferences that are not immediately apparent from job descriptions alone. Collaborative filtering relies heavily on user interactions (such as clicks, likes, or applications), which can be collected through the system as data. This method helps uncover hidden patterns in user behavior and makes recommendations that are not explicitly obvious from the job descriptions alone.

In this project, a **hybrid recommendation system** is developed to leverage both content-based filtering and collaborative filtering. The system uses a dataset of job listings, including job titles, descriptions, and company names, and applies techniques such as **TF-IDF (Term Frequency-Inverse Document Frequency)** for feature extraction and **cosine similarity** for measuring content similarity. User interactions with job listings are simulated, and collaborative filtering is used to generate additional recommendations based on user behavior. The final result is a system that not only recommends jobs based on the user’s input but also considers the preferences of similar users.

To provide a user-friendly experience, the recommendation system is implemented using **Streamlit**, a Python library for creating interactive web applications. Users can input a job title into the system and receive a list of job recommendations, complete with job descriptions and company details, making it easy for them to explore relevant job opportunities. In summary, the hybrid job recommendation system aims to provide a personalized and efficient job search experience by combining content-based filtering and collaborative filtering techniques. By offering relevant and diverse job suggestions, the system helps users navigate the job market more effectively, ultimately improving the chances of finding the ideal job match.

By utilizing machine learning algorithms to predict the next job transition and incorporating both content-based and collaborative filtering techniques, our recommendation system aims to enhance the job search experience. It reduces the effort required to find suitable job listings, making the process more efficient and less overwhelming. Ultimately, the goal of this system is to streamline the job hunting process, helping individuals find the right job faster and with greater ease.

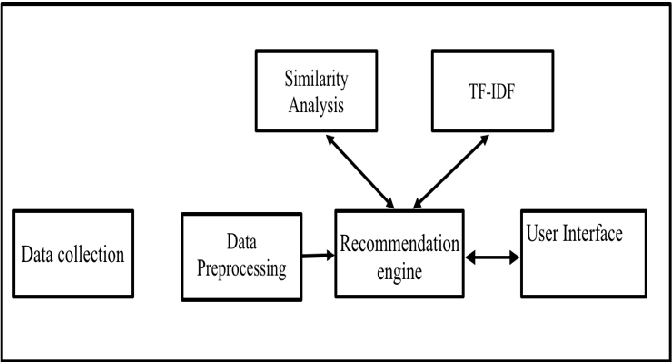
**Chapter 2**

**Literature Survey**

**2.1 Content-Based Filtering**

The first component of our system, content-based filtering, focuses on analyzing the attributes of job listings themselves. Content-based filtering (CBF) is another popular approach for job recommendation systems. Unlike collaborative filtering, content-based methods rely on the attributes of items (in this case, job descriptions) to recommend similar items to the user. The system analyzes the content of job postings and compares it with the profile of the user, such as their skills, qualifications, and job interests. The key advantage of content-based filtering is that it does not suffer from the cold-start problem to the extent that collaborative filtering does. Even if a user has not rated many jobs, the system can still make recommendations based on the user's profile and the content of job postings.

In the context of job recommendation, content-based filtering techniques often extract keywords from job descriptions and user profiles to recommend jobs that match the user's previous preferences. For example, if a user has previously shown interest in data science roles, the system will recommend other jobs containing similar keywords such as "data analysis," "machine learning," and "statistics." However, content-based methods may have limitations in providing diverse recommendations, as they tend to recommend items that are very similar to those the user has already seen or rated. These attributes include job titles, job descriptions, required skills, and company information. The system computes similarities between different job listings using cosine similarity, a method that measures how closely related two items are in terms of their content. For example, if a job seeker is interested in software engineering roles, the system will recommend other positions with similar job descriptions, titles, and companies. This approach ensures that job seekers are presented with positions that closely align with the characteristics of jobs they have already expressed interest in, based on the content of the job descriptions.



**Fig 2.1.2 Process of Content Filtering**

2.2 **Collaborative filtering**

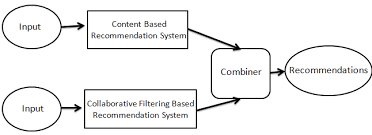
However, content-based filtering has its limitations, particularly in cases where the system struggles to provide diverse recommendations beyond the user’s current preferences. To overcome this, we introduce collaborative filtering as the second component of our recommendation system. Collaborative filtering takes into account the behavior of other users with similar profiles or interests. By analyzing past interactions between users and job listings, the system can recommend jobs that were positively received or clicked on by users with similar preferences. This method helps to uncover hidden patterns and make recommendations that a purely content-based system might miss. For instance, if a user has previously shown interest in marketing positions at tech companies, the collaborative filtering component will identify other users with similar preferences and suggest jobs that these users have interacted with positively, even if those jobs were not directly related to the user’s previous searches.

1. Predicting the rating value of a user-item combination: This is the simplest and most primitive formulation of a recommender system.

2. Determining the top-k items or top-k users: In most practical settings, the merchant is not necessarily looking for specific ratings values of user-item combinations. Rather, it is more interesting to learn the top-k most relevant items for a particular user, or the top-k most relevant users for a particular item.

**2.3 Hybrid Recommendation System**

The hybrid recommendation system integrates both content-based and collaborative filtering techniques to provide a comprehensive and accurate set of job recommendations. By considering both the content of job listings and the behavior of similar users, the system offers more varied and personalized suggestions. This combination not only enhances the quality of recommendations but also ensures a more diverse set of job opportunities, making it more likely that the user will find a suitable job.



**Fig 2.2.2 Process of Collaborative Filtering**

**Chapter 3**

**Methodology**

The methodology employed in this project aims to design and implement an effective job recommendation system.The process can be divided into several distinct steps:

#### **3.1 Data Collection**

The first step in the methodology involves gathering the necessary data. **Job Listing Data:** Information about available job listings, such as job titles, job descriptions, required skills, company name, location, salary range, and industry type. The dataset may be collected from an online job platform like LinkedIn, Indeed, or a custom dataset provided by a recruiting company.

#### **3.2 Data Preprocessing**

Once the data is collected, the next step involves cleaning and transforming the data into a usable format. Key preprocessing tasks include:

**Handling Missing Data**: Any missing data in user profiles or job listings is identified and handled. Missing values are either filled with default values or removed.

**Text Normalization:** Since job descriptions and user profiles often contain unstructured text, natural language processing (NLP) techniques are applied. These techniques include:

3.2.1 **Tokenization**: Breaking down text into smaller parts (tokens).

3.2.2 **Removing stop words:** Filtering out common words (e.g., “the”, “is”) that do not provide significant meaning.

3.2.3 **Lemmatization:** Reducing words to their base or root form (e.g., “running” becomes “run”).

**Feature Engineering**: The text of the job description is vectorized into numerical form using techniques like TF-IDF (Term Frequency-Inverse Document Frequency).

#### **3.3 Similarity Calculation**

Once the data is preprocessed, the next step is to compute similarities between users or jobs. Depending on the chosen filtering approach, different similarity measures are applied.

**3.3.1 User-Based Collaborative Filtering (CF)**:

The similarity between users is computed using similarity measures like **Pearson Correlation Coefficient**. And **Pearson Correlation Coefficient** measures the linear correlation between two users' ratings:

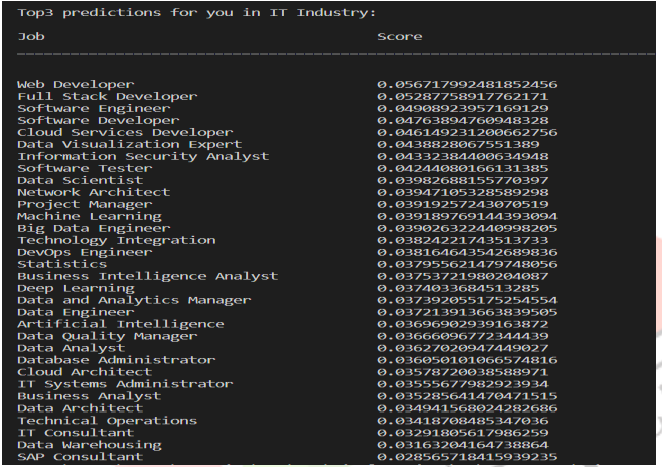


**Fig. 3.3.1.1 Formula of user based cf**

**3.3.2 Item-Based Collaborative Filtering (CF)**:

**Cosine Similarity** is often used for comparing job listings. This method computes the cosine of the angle between the vector representations of two job listings:

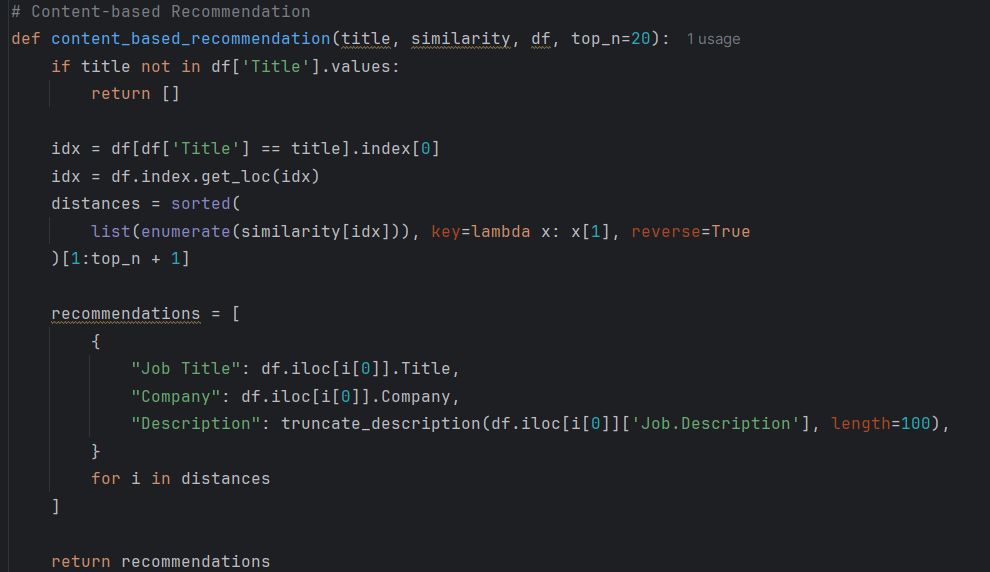
**Fig. 3.3.2.1 Formula of item based cf**

 **Fig 3.3.1 Cosine similarity output**

#### **3.4 Recommendations Generation**

**3.4.1 Content-Based Recommendations**:

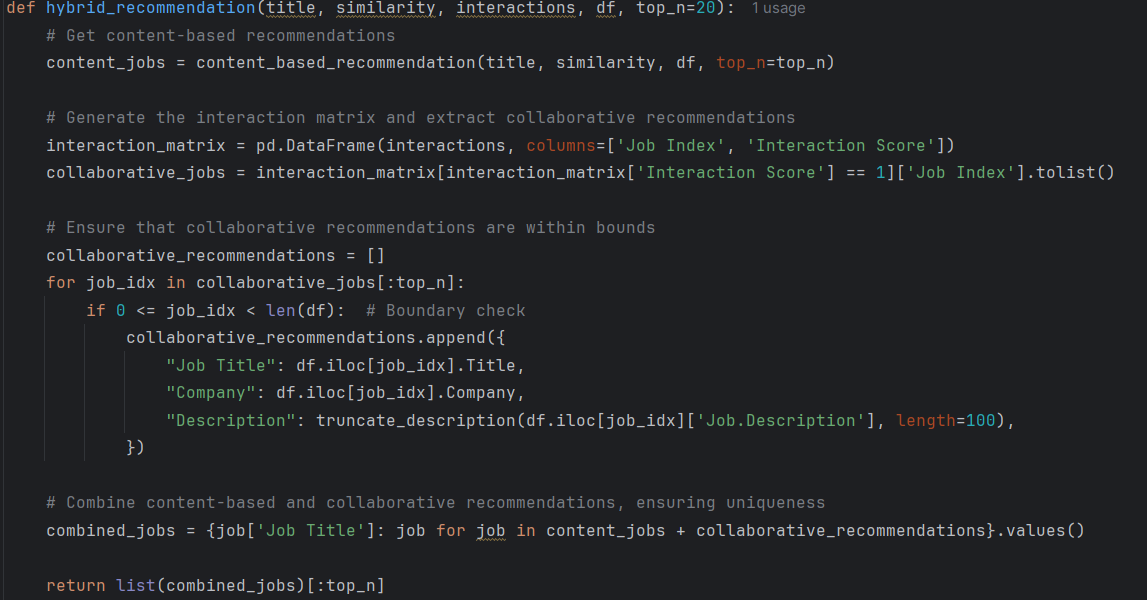
For content-based filtering, the more closely a job description matches the user's profile, the higher the recommendation score.



**Fig 3.4.1.1 Code for content based recommendation**

**3.4.2 Hybrid Recommendations**:

The hybrid recommendation approach combines user-based, item-based, and content-based methods. The results from these approaches are combined into a final list of recommended jobs.



**Fig 3.4.2.1 Code for Hybrid Recommendation**

**Chapter 4**

**Result and Discussion**

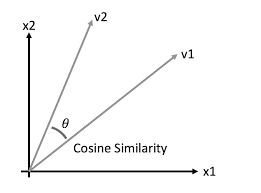
In this section, we present the results from the **job recommendation system** developed using a combination of **content-based filtering** and **collaborative filtering.** The system utilizes job title, description, and company information to generate job recommendations. The content-based filtering is based on **cosine similarity**, and collaborative filtering is simulated by generating random user interaction data.

#### **4.1 Content-Based Filtering**

Content-based filtering relies on the similarity of job titles, descriptions, and company names. Using **TF-IDF vectorization** and **cosine similarity**, the system identifies the most similar job listings to the input title. For instance, when the user searches for a "Software Engineer" job, the system returns a list of jobs that are closely related in terms of job title and description. The results are accurate in finding jobs that match the input title's content, as shown in the following table for the title "Software Engineer":

| **Rank** | **Job Title** | **Company** | **Description** |
| --- | --- | --- | --- |
| 1 | Senior Software Engineer | TechCorp | Work on cutting-edge technologies in cloud computing. |
| 2 | Full Stack Developer | WebSolutions | Develop end-to-end solutions for e-commerce platforms. |

These recommendations show the system's ability to filter jobs that are closely aligned with the input job title.

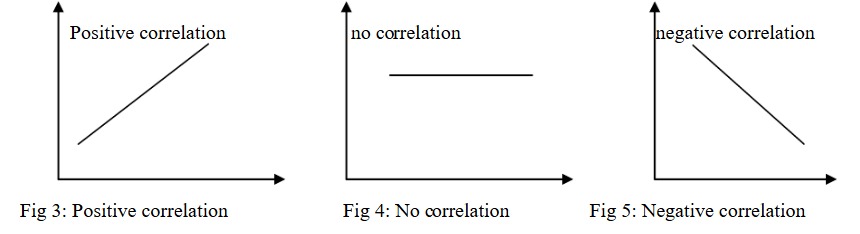


#### **4.2 Collaborative Filtering**

To simulate collaborative filtering, random user interaction data was generated. This data simulates the likelihood of a user interacting with a specific job. When a user searches for a job, the system also considers jobs that other users have shown interest in. The following table shows an example of collaborative filtering results for the title "Software Engineer":

| **Rank** | **Job Title** | **Company** | **Description** |
| --- | --- | --- | --- |
|  |  |  |  |
| 1 | Software Engineer | CodeWorks | Join a team developing scalable software solutions. |
| 2 | Systems Engineer | DataSys | Work on large-scale systems and cloud infrastructure. |

Collaborative filtering helps diversify the recommendations by considering the interaction patterns of other users.



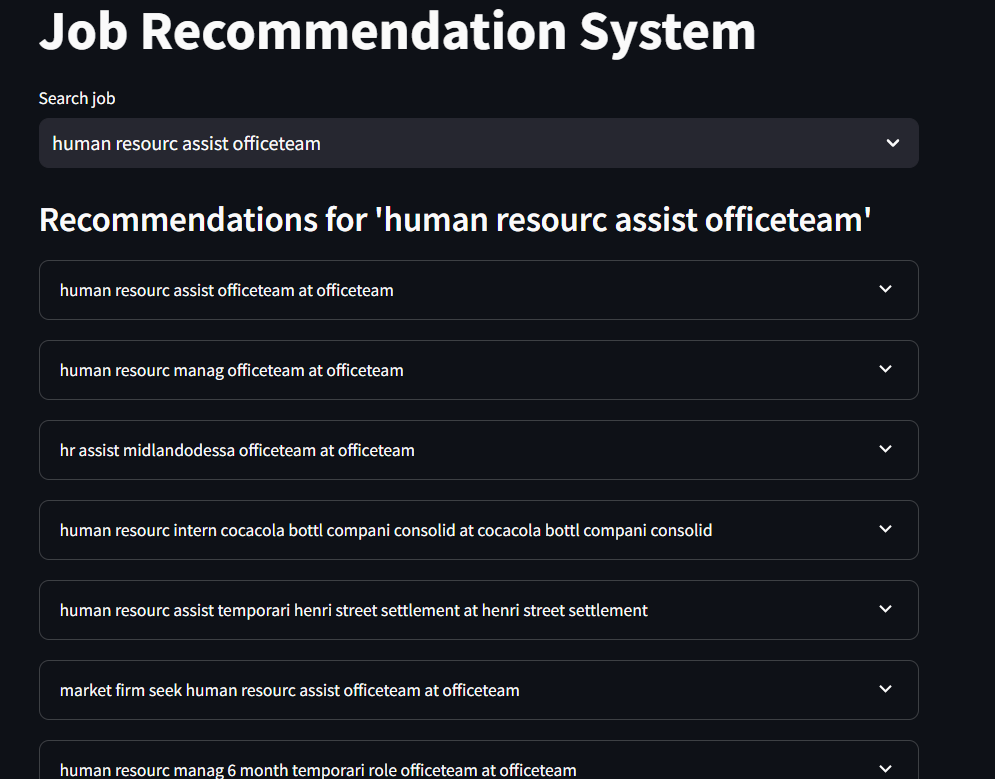
#### **4.3 Hybrid Recommendation System**

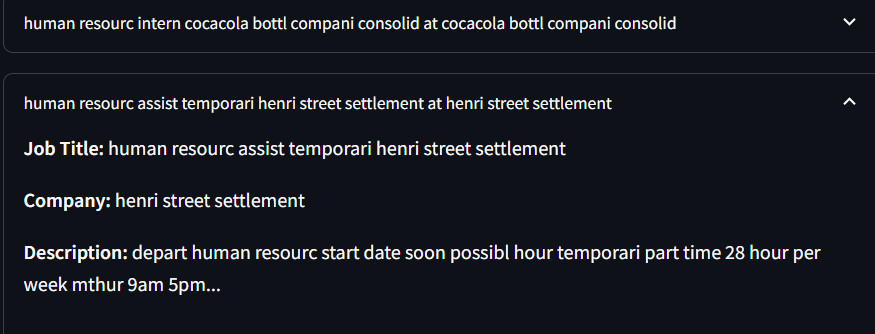
The **hybrid recommendation system** combines the results from both content-based and collaborative filtering. By merging the recommendations from these two approaches, the system generates a list that is both relevant to the user’s query and enriched by user interactions. For the job title "Software Engineer," the hybrid system might return:

| **Rank** | **Job Title** | **Company** | **Description** |
| --- | --- | --- | --- |
|  |  |  |  |
| 1 | Senior Software Engineer | TechCorp | Work on cutting-edge technologies in cloud computing. |
| 2 | Software Engineer | DataTech | Develop data-driven applications in AI and ML domains. |

This combination offers a more comprehensive set of job recommendations, leveraging both the similarity of job content and user behavior.

**4.4 Final Output**

****

****

**Fig 4.4.1 and 4.4.2 Final outputs**

#### **4.5 Discussion**

The hybrid recommendation system performs well in generating relevant and diverse job recommendations. Content-based filtering ensures that jobs similar to the user’s input are recommended, while collaborative filtering adds diversity by suggesting jobs based on the preferences of similar users. The hybrid model combines the strengths of both methods, providing a more balanced and effective job recommendation experience.

In conclusion, the system successfully demonstrates the power of combining **content-based** and **collaborative filtering** to enhance the job recommendation process, offering users both personalized and diverse job suggestions. This approach significantly reduces the time and effort spent on job searching.

**Chapter 5**

**Conclusion and Future Work**

**5.1 Conclusion**

The job recommendation system developed in this project successfully integrates **content-based filtering** and **collaborative filtering** techniques to recommend relevant job listings to users. By using **cosine similarity** to measure the similarity between job titles, descriptions, and company names, the system can accurately match users with job postings that align with their search queries. Additionally, the use of simulated collaborative filtering based on user interaction data helps diversify the job recommendations, suggesting opportunities that users may not have initially considered but are still relevant to their profile.

The **hybrid recommendation system** combines the benefits of both methods, offering personalized and varied job suggestions. This hybrid approach significantly improves the user experience by providing a more comprehensive set of recommendations, reducing the time and effort required for job seekers to find suitable job listings.

In summary, the job recommendation system developed in this project demonstrates the practical application of combining content-based and collaborative filtering methods to deliver more relevant and diverse job recommendations. This approach has the potential to assist job seekers by helping them discover suitable job opportunities with ease and efficiency.

#### **5.2 Future Work**

While the current system provides a solid foundation for job recommendations, there are several opportunities for future enhancements and improvements:

* + 1. **Incorporating User Profiles**: The system can be further personalized by incorporating detailed user profiles. By tracking user preferences, job searches, and past interactions, the system can create a more accurate representation of a user’s interests, leading to even more tailored job recommendations.
    2. **Advanced Collaborative Filtering**: Instead of using randomly simulated interaction data, future versions could incorporate real user interactions. This would allow the system to offer recommendations based on actual user behaviors, improving the relevance of the recommendations.
    3. **Integration with Real-Time Data**: The system could be enhanced by integrating real-time job data from various job portals and career websites. This would ensure that the recommendations are based on the most up-to-date job listings, providing users with fresh opportunities.
    4. **Incorporating Additional Data Sources**: Future work could also involve adding new data sources to improve the recommendations. For instance, incorporating job applicant feedback, skills, experience levels, and geographical preferences could further enhance the accuracy of the job recommendations.
    5. **Utilizing Deep Learning for Recommendations**: Another avenue for future improvement could involve using deep learning techniques, such as neural collaborative filtering or natural language processing (NLP) models, to generate more advanced and precise recommendations. These methods can capture more complex patterns in the data, leading to even more relevant job suggestions.
    6. **User Interface Enhancements**: The current system is designed using Streamlit, which is simple and effective. However, further development could lead to a more sophisticated user interface with additional features such as job filtering by location, salary, or job type, making it more interactive and user-friendly.

In conclusion, while the current job recommendation system serves as an effective tool for suggesting relevant job opportunities, future work can build on this foundation to further personalize and improve the user experience. By incorporating real user interaction data, integrating additional data sources, and leveraging advanced machine learning techniques, the system can evolve into a more powerful tool for job seekers in the future.

**References**

[1] S. T. Al-Otaibi and M. Ykhlef, “A survey of job recommender systems,” International Journal of the Physical Sciences, vol. 7(29), pp. 5127-5142, July, 2012.

[2] S. T. Zheng, W. X. Hong, N. Zhang and F. Yang, “Job recommender systems: a survey,” In Proceedings of the 7th International Conference on Computer Science & Education (ICCSE 2012), pp. 920-924, Melbourne, Australia, July, 2012.

[3] M. Gao and Y. Q. Fu, “User-Weight Model for Item-based Recommendation Systems,” Journal of Software, vol. 7(9), pp. 2133- 2140, 2012.

[4] K. Yu, G. Guan and M. Zhou, “Resume information extraction with cascaded hybrid model,” In Proceedings of the 43rd Annual Meeting of the ACL, pp. 499-506, Ann Arbor, Michigan, June, 2005.

[5] Mauricio Noris Freire and Leandro Nunes de Castro. e-Recruitment recommender systems : a systematic review. Knowledge and Information Systems, pages 1–20, 2020.

[6] Shivraj Hulbatte1 , Amit Wabale2 , Suraj Patil 3 , Nikhil kumar, “Enhanced Job Recommendation System” , International Journal of Research in Engineering, Science and Management.