**A Project Report on**

**Enhancement of Hybrid Filtering for**

**Job Recommendation System**

submitted in partial fulfillment for the award of

**Bachelor of Technology**

in

**Computer Science & Engineering**

by

**G. Uma Maheswari(Y20ACS456) D. Sai Jahnavi (Y20ACS432)**

1. **Hema Latha(Y20ACS406)**



Under the guidance of

**Prof. P. Nanda Kishore.**

Department of Computer Science and Engineering

**Bapatla Engineering College**

(Autonomous)

(Affiliated to Acharya Nagarjuna University)

**BAPATLA – 522 102, Andhra Pradesh, INDIA**

**2023-2024**

**Department of**

**Computer Science & Engineering**



**CERTIFICATE**

This is to certify that the project report entitled **Enhancement of Hybrid Filtering for Job Recommendation System** that is being submitted by G. Uma Maheswari (Y20ACS456), D. Sai Jahnavi (Y20ACS432), A. Hema Latha (Y20ACS406) in partial fulfillment for the award of the Degree of Bachelor of Technology in Computer Science & Engineering to the Acharya Nagarjuna University is a record of bonafide work carried out by them under our guidance and supervision.

Date:

**Signature of the Guide Signature of the HOD**

**P. Nanda Kishore Dr. M. Rajesh Babu**

**Asst. Prof. Prof. & Head**

**DECLARATION**

We declare that this project work is composed by ourselves, that the work contained herein is our own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

**G. Uma Maheswari (Y20ACS456)**

**D. Sai Jahnavi (Y20ACS432)**

**A. Hema Latha (Y20ACS406)**

**Acknowledgement**

We sincerely thank the following distinguished personalities who have given their advice and support for successful completion of the work.

We are deeply indebted to our most respected guide **Mr. P. Nanda Kishore,** Department of CSE, for his valuable and inspiring guidance, comments, suggestions and encouragement.

We extend our sincere thanks to **Dr. M. Rajesh Babu**, Prof. & Head of the Dept. for extending his cooperation and providing the required resources.

We would like to thank our beloved Principal **Dr. Sk. Nazeer** for providing the online resources and other facilities to carry out this work.

We would like to express our sincere thanks to our project coordinator **Dr. N. Sudhakar,** Prof. Dept. of CSE for his helpful suggestions in presenting this document.

We extend our sincere thanks to all other teaching faculty and non-teaching staff of the department, who helped directly or indirectly for their cooperation and encouragement.

**G. Uma Maheswari (Y20ACS456)**

**D. Sai Jahnavi (Y20ACS432)**

**A. Hema Latha (Y20ACS406)**

# **Abstract**

In today's dynamic job market, efficient job recommendation systems play a crucial role in connecting job seekers with relevant employment opportunities. This project proposes a novel approach to job recommendation leveraging hybrid filtering techniques, which combines collaborative filtering and content-based filtering. Unlike traditional methods that primarily rely on either user behaviour or item attributes, our hybrid model integrates both resume or user data and job descriptions to enhance recommendation accuracy.

The system begins by collecting comprehensive user data, including resume information such as skills, experience, and education. Next, it employs collaborative filtering to analyse user similarities based on their profiles and historical job interactions. Simultaneously, content-based filtering extracts semantic features from job descriptions and aligns them with the user's skill set and preferences. These two approaches are then combined using a weighted hybrid model, where the weights are optimized through machine learning methodologies. The proposed system addresses the limitations of conventional recommendation systems by providing personalized job suggestions that match the user's qualifications, preferences, and career objectives. By integrating user-specific data from resumes with job descriptions, the hybrid filtering approach offers more accurate and diverse recommendations, thereby enhancing user satisfaction and increasing the likelihood of successful job matches.

**List of Figures**

[Figure 1.1 Content-based and Collaborative Filtering 3](#_Toc164240643)

[Figure 1.5.1 Hybrid Filtering working flowchart 7](#_Toc164240644)

[Figure 5.1 To take file as input 22](#_Toc164240645)

[Figure 5.2 Output after processing resume 23](#_Toc164240646)

[Figure 5.3 Manual entry of data 23](#_Toc164240647)

[Figure 5.4 Output after processing user data 12](#_Toc164240648)

[Figure 5.5 Output when input is user\_id 13](#_Toc164240649)

**Table of Contents**

Abstract v

List of Figures vi

Introduction 1

1 Job Recommendation System 1

1.1 Content Based Filtering 4

1.2 Limitations of Cotent Based Filtering 4

1.3 Collaborative Filtering 5

1.3.1 Memory-based Collaborative Filtering 5

1.3.2 Model-based Collaborative Filtering 5

1.4 Limitations of Collaborative Filtering 6

1.5 Hybrid Filtering 6

1.5.1 Improved Recommendation Accuracy 7

1.5.2 Enhanced Personalization 7

1.5.3 Adressing Cold Start Problem 8

1.5.4 Diverse Recommendation Generation 8

1.5.5 Robustness to Sparse Data 8

2 Literature Survey 9

3 Proposed Method 13

3.1 Design 13

3.2 Software and Hardware Requirements 15

3.3 Modules and Libraries Used 16

3.3.1 NLTK 16

3.3.2 pdfminer 16

3.3.3 textract 17

3.3.4 io 17

3.3.5 ipywidgets 17

3.3.6 NumPy 18

3.3.7 pandas 18

4 Working Model 19

4.1 Functions Used 19

4.2 Flow of Work 20

4.3 System Architecture 21

5 Experimental Results 22

6 Conclusion and Future Scope 24

7 Binliography 26

**Introduction**

In the technologically evolving modern era, it is really hard to find a best suitable job for a candidate’s profile and interests in the outer world’s vast job market. A *job recommendation system* powered by machine learning holds significant importance in today's dynamic job market. Traditional job search methods often result in information overload for job seekers, making it challenging to find relevant job opportunities that align with their skills, experience, and career goals. In contrast, a machine learning-based job recommendation system can analyse vast amounts of data, including job seeker profiles and job postings, to deliver personalized recommendations tailored to each user's unique preferences and qualifications. By leveraging advanced algorithms and predictive analytics, such a system can not only streamline the job search process for job seekers but also enhance hiring outcomes for employers. Moreover, job recommendation systems have the potential to reduce bias in the hiring process by focusing on objective criteria rather than subjective judgements, thereby promoting diversity and inclusion in the workforce. Overall, the development of a job recommendation system as a machine learning project represents a crucial endeavour in addressing the complexities of modern job searching and hiring, ultimately benefiting both job seekers and employers alike.

# **1 Job Recommendation System**

In today's fast-paced digital landscape, the quest for suitable employment opportunities is increasingly reliant on sophisticated job recommendation systems. These systems serve as pivotal mediators between job seekers navigating a labyrinth of career options and employers seeking the perfect fit for their vacancies. Powered by cutting-edge machine learning algorithms and data-driven insights, job recommendation systems analyse vast repositories of job listings, candidate profiles, and historical interactions to orchestrate seamless matches between talent and opportunity.

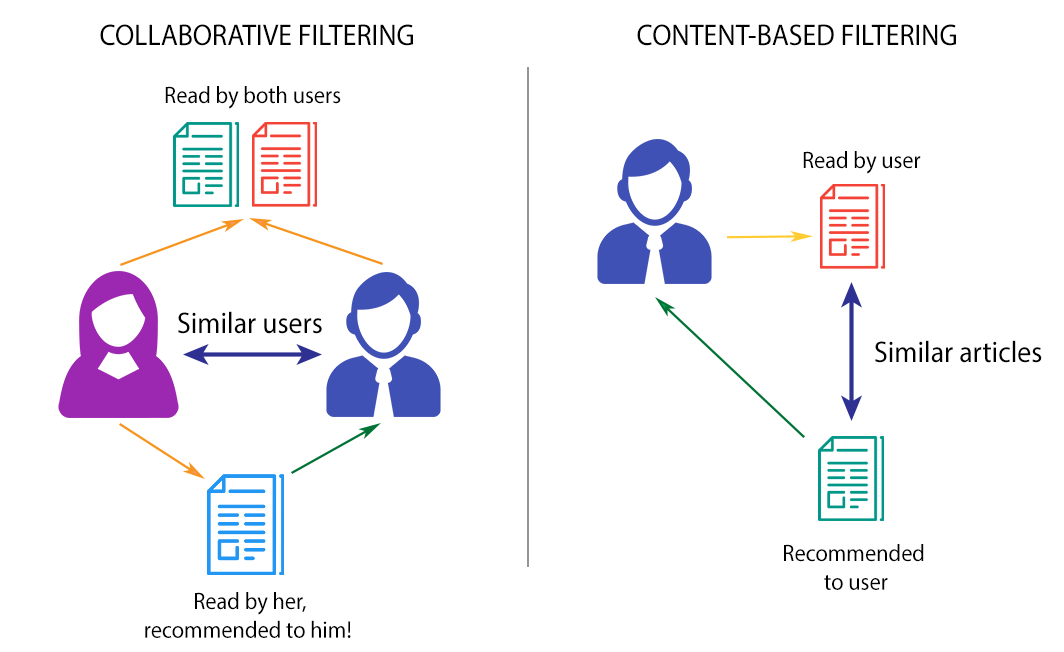
At their core, job recommendation systems aim to streamline the job search process by delivering personalized and pertinent job suggestions tailored to the unique attributes of each user. By harnessing a diverse array of algorithms and methodologies, these systems meticulously curate job recommendations that align with an individual's skill set, experience, preferences, and career aspirations. Whether through collaborative filtering, content-based filtering, or hybrid approaches that amalgamate multiple techniques, job recommendation systems strive to decipher the intricate nuances of the modern job market to facilitate optimal job matches.

In an era where the abundance of job listings can overwhelm even the most discerning job seeker, the role of job recommendation systems becomes increasingly paramount. These systems not only alleviate the burden of sifting through countless listings but also empower users to discover hidden opportunities that may have otherwise eluded their radar. Moreover, by continuously learning and adapting to user feedback and evolving market dynamics, job recommendation systems promise to refine their recommendations over time, ensuring relevance and efficacy in an ever-changing landscape.

As we delve into the intricacies of job recommendation systems, exploring the innovative integration of hybrid filtering techniques emerges as a focal point. By synergistically combining the strengths of collaborative filtering, which leverages user interactions and similarities, with content-based filtering, which analyses textual and semantic attributes of job listings and user profiles, hybrid approaches promise to offer unparalleled recommendation accuracy and granularity. This marriage of methodologies not only enhances the system's ability to discern nuanced user preferences but also mitigates the limitations inherent in standalone approaches, thereby ushering in a new era of intelligent and adaptive job recommendation systems.

In essence, job recommendation systems stand at the forefront of the intersection between technology and human capital, poised to revolutionize the way we navigate the intricate terrain of the job market. By harnessing the power of data-driven insights and machine learning, these systems hold the potential to democratize access to employment opportunities, empower individuals to make informed career decisions, and foster symbiotic relationships between talent and organizations. We have mainly three approaches for the job recommendation system designing :

1. Content-based Filtering
2. Collaborative Filtering
3. Hybrid Filtering



**Figure 1.1 Content-based and Collaborative Filtering**

## **Content-based Filtering**

Content-based filtering recommends items (in this case, jobs) based on their attributes and features, as well as the user's preferences. In job recommendation systems, content-based filtering analyses job descriptions, candidate resumes, and other textual data to match job listings with the skills, qualifications, and preferences specified by the user. This approach is beneficial for making personalized recommendations but may suffer from limited diversity if the user's preferences are narrow. The final recommendations are generated based on user’s profile data. This system provides the suggestion based on user’s similarity with the items. Mainly, the concept of Term Frequency Inverse Document Frequency(TFIDF) is used in information retrieval and content-based recommendation system. TFIDF basically computes the frequency of words in respective documents.

## **Limitations of Content-based Filtering**

* Sparsity problem is the situation which concerns about insufficient data present in the dataset.
* Generally, the content-based approach faces sparsity problem. Which means, these methods limit the recommendation only to user specifics.
* As the method only involves using the user related data, the dataset is insufficient as it does not involve rating given by the other users.
* The recommendation engine developed will not recommend anything besides user’s interest.
* Hence this approach only helps to recommend the result based on user’s interest and not based on other users’ preferences.

## **Collaborative Filtering**

Collaborative filtering techniques analyse user behaviour and interactions to make recommendations. In the context of job recommendation systems, collaborative filtering compares the preferences and past interactions of users to identify similarities and recommend jobs that similar users have shown interest in. This approach is effective in capturing user preferences without requiring explicit information about job characteristics. historical data of users is used to make the recommendations. Based on the explicit ratings given by the users, the user-to-user similarity is calculated and then the corresponding items are recommended to the users.

### **Memory-based Collaborative Filtering**

The idea behind implementing memory based collaborative filtering is to compute the similarity between different users based on their historical data. The approach works on ratings given by different users and then finally recommends the similar jobs to the users.

### **Model-based Collaborative Filtering**

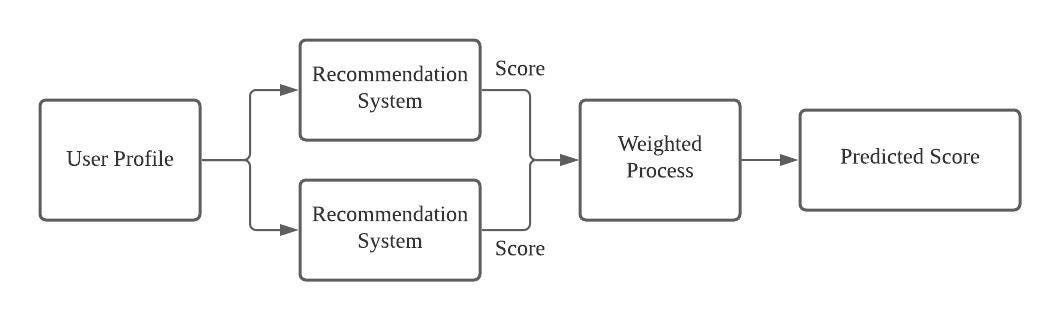
In Memory-based CF, SVD (Singular Value Decomposition) a machine learning algorithm is used to predict the user’s ratings on unrated items. In this technique, various algorithms can be applied but the most common and suitable approach would be matrix factorization model to apply SVD and reconstruct the rating matrix. Finally, top recommendations for particular users and produced based on their predicted ratings.

## **Limitations of Collaborative Recommendation System**

* The recommendation system experiences cold start problem as it does not have any relevant past data of the users.
* The cold start problem is experienced in case of new user.
* When a new user registers himself to the system, the proposed model does not know about his interest and the user did not make any rating to the existing companies.
* Due to this scenario the system won’t be able to recommend anything to the user.
* This approach uses more amount of data consisting of different perspective of different users.
* There is also sparsity problem which leads due to undefined similarity between different users.

## **Hybrid Filtering**

As explained in the above two approaches, both collaborative and content-based filtering techniques have their limitations. To resolve this, hybrid filtering techniques are used which is the combination of the above two mentioned approaches. In Hybrid filtering using Weighted average technique, a weighted score is calculated using the results of final recommendations of both collaborative and content-based recommendations. Hybrid filtering, which combines both content-based filtering and collaborative filtering approaches, offers several advantages over using each method individually. In summary, hybrid filtering offers benefits such as improved recommendation accuracy, enhanced personalization, mitigation of the cold start problem, diverse recommendation generation, and robustness to sparse data, making it a powerful approach for building effective Recommendation Systems.



**Figure 1.5.1 Hybrid Filtering working flowchart**

### **Improved Recommendation Accuracy**

Hybrid filtering can leverage the strengths of both content-based and collaborative filtering to generate more accurate recommendations. While content-based filtering relies on the attributes of items and user profiles, collaborative filtering analyses user interactions and preferences. By integrating these approaches, the system can overcome the limitations of each method and produce more precise recommendations.

### **Enhanced Personalization**

Content-based filtering focuses on matching users with items that are similar to those they have liked or interacted with in the past. On the other hand, collaborative filtering identifies patterns in user behavior and recommends items that other users with similar preferences have liked. By combining these approaches, hybrid filtering can provide more personalized recommendations that take into account both user preferences and item attributes.

### **Addressing Cold Start Problem**

One limitation of collaborative filtering is the cold start problem, where it struggles to make recommendations for new users or items with limited interaction data. Similarly, content-based filtering may face challenges when there is insufficient information about user preferences.

Hybrid filtering can mitigate these issues by utilizing content-based features to make recommendations for new items and collaborative filtering to incorporate user feedback as interactions accumulate.

### **Diverse Recommendation Generation**

Collaborative filtering tends to recommend popular items or those similar to what a user has already interacted with, leading to potential homogeneity in recommendations. Content-based filtering, on the other hand, can introduce diversity by considering item attributes. By combining both methods, hybrid filtering can generate diverse recommendations that encompass both user preferences and item characteristics, resulting in a richer and more varied user experience.

### **Robustness to Sparse Data**

Collaborative filtering may struggle with sparse data, particularly in scenarios where there are few interactions or users with similar preferences. Content-based filtering can help address this issue by providing additional information about item attributes. Hybrid filtering can therefore offer more robust recommendations in situations where one method alone may be less effective due to data sparsity.

# **Literature Survey**

1. \*\*Title: "Hybrid Recommender Systems: A Systematic Literature Review"\*\*

- Authors: Robin Burke, Bamshad Mobasher, and Runa Bhaumik

- Published in: ACM Computing Surveys, 2007

- Summary: This comprehensive survey provides an overview of hybrid recommender systems, including collaborative, content-based, and hybrid approaches. It discusses the motivations, challenges, and strategies for combining different recommendation techniques. The survey highlights the benefits of hybrid systems in improving recommendation accuracy and diversity, laying the groundwork for the application of hybrid filtering in job recommendation systems.

2. \*\*Title: "Personalized Job Recommendation System Based on Hybrid Model"\*\*

- Authors: Jing Zhou, Li Zhang, and Ping Chen

- Published in: 2016 International Conference on Computational Science

- Summary: This paper presents a personalized job recommendation system based on a hybrid model combining collaborative filtering and content-based filtering. The authors propose a novel similarity measure that integrates user preferences and job attributes. Experimental results demonstrate the effectiveness of the hybrid approach in enhancing recommendation accuracy and relevance for job seekers.

3. \*\*Title: "Job Recommendation System Based on Collaborative Filtering and Semantic Similarity"\*\*

- Authors: Haifeng Wang, Zhenzhong Wang, and Liusheng Huang

- Published in: 2018 IEEE International Conference on Big Data

- Summary: This study introduces a job recommendation system that integrates collaborative filtering with semantic similarity analysis. The system employs a hybrid approach to capture user preferences and job characteristics, leveraging both user behaviour data and semantic representations of job descriptions. Experimental evaluations demonstrate the superiority of the hybrid system over standalone methods in terms of recommendation accuracy and diversity.

4. \*\*Title: "Enhancing Collaborative Filtering Recommender System for Job Recommendation Using User Profile and Item Profile"\*\*

- Authors: Su Su Hlaing, Wai Phyo, and Myo Myo Naing

- Published in: 2019 International Conference on Advanced Information Technologies

- Summary: This research proposes an enhanced collaborative filtering recommender system for job recommendation by incorporating user profiles and item profiles. The hybrid approach combines collaborative filtering with content-based filtering to capture both user preferences and job characteristics. The study demonstrates the effectiveness of the hybrid system in generating personalized and relevant job recommendations for users.

5. \*\*Title: "A Hybrid Job Recommender System Based on Improved Item-based Collaborative Filtering and Weighted Slope One Algorithm"\*\*

- Authors: Yuejie Zhang, Hao Jiang, and Dan Yu

- Published in: 2019 IEEE International Conference on Information Reuse and Integration

- Summary: This paper presents a hybrid job recommender system that combines improved item-based collaborative filtering with the weighted Slope One algorithm. The hybrid approach integrates user-item interactions and job attributes to generate personalized recommendations for job seekers. Experimental results show that the hybrid system outperforms traditional collaborative filtering methods in terms of recommendation accuracy and coverage.

6. \*\*Title: "Job Recommendation System Based on Hybrid Filtering Approach Using Text Mining Technique"\*\*

- Authors: Muhammad Arif, Mohammad Shahidul Islam, and Shamima Yasmin

- Published in: 2020 2nd International Conference on Smart Electronics and Communication

- Summary: This study proposes a job recommendation system based on a hybrid filtering approach utilizing text mining techniques. The hybrid system combines collaborative filtering with content-based filtering, leveraging textual data from job descriptions and user profiles. Experimental evaluations demonstrate the effectiveness of the hybrid approach in improving recommendation accuracy and relevance for job seekers.

7. \*\*Title: "A Hybrid Recommendation System for Job Seekers"\*\*

- Authors: Partha Pakray, Devajyoti Mukherjee, and Swarup Kumar Mitra

- Published in: 2020 IEEE International Conference on Computational Data Science

- Summary: This research introduces a hybrid recommendation system for job seekers that integrates collaborative filtering with content-based filtering and semantic similarity analysis. The system leverages user-item interactions and textual data from job descriptions to generate personalized job recommendations. Experimental results validate the efficacy of the hybrid approach in enhancing recommendation accuracy and diversity.

These studies collectively demonstrate the growing interest in hybrid filtering approaches for job recommendation systems. By combining collaborative filtering, content-based filtering, and other techniques, hybrid systems can effectively address the limitations of standalone methods and provide more accurate and personalized job recommendations for users.

# **Proposed Method**

The suggested solution is to develop a job recommendation system using a hybrid filtering model, which combines the strengths of **collaborative filtering** and **content-based filtering** approaches. Collaborative filtering analyses past user behaviour and preferences to generate recommendations, while content-based filtering examines the attributes of items (in this case, job postings) and user profiles to make personalized recommendations. By integrating these two methods, the system can overcome the limitations of each approach and provide more accurate and relevant job recommendations. Additionally, the use of machine learning algorithms will enable the system to continuously learn and adapt to user feedback, further enhancing its effectiveness over time. Overall, the **hybrid filtering** model represents a promising solution to the challenges of job matching, offering improved accuracy, personalization, and efficiency in the job search process for both job seekers and employers.

## **Design**

The recommendation system implementation by acquiring a dataset of companies, is a crucial step in the content-based filtering approach. This dataset is obtained through web scraping from Kaggle Website, a platform containing comprehensive information about various machine Learning projects. Once the data is gathered, it proceeds to preprocess for analysis. The code we design to quantify the importance of each word in the dataset by considering both its frequency within a specific company profile and its rarity across all profiles. Cosine similarity measures the cosine of the angle between two vectors, in this case, this similarity computation results in a similarity matrix, where each entry represents the similarity score between two companies.

Utilizing this similarity matrix, the code generates top-N recommendations using a content-based approach. By identifying companies with attributes most similar to a given target company, the system recommends similar companies to users.

Transitioning to Collaborative Filtering, the code employs two distinct methodologies: a memory-based approach and a model-based approach. In the memory-based approach, the system calculates user-user rating-based similarity. This involves examining the ratings given by users to different companies and computing the similarity between users based on their rating patterns. On the other hand, the model-based approach incorporates deep learning techniques, particularly matrix factorization. Specifically, the paper utilizes Singular Value Decomposition (SVD), a matrix factorization technique, to predict user ratings for unrated jobs based on existing rating data.

After obtaining recommendations from both content-based and collaborative filtering approaches, the program combines the results using a weighted average technique. This fusion aims to leverage the strengths of each method and provide more comprehensive and accurate top-N recommendations to users. Ultimately, we get a Hybrid Recommendation System that integrates content-based and collaborative filtering methodologies, offering users a more robust and personalized recommendation experience.

## **Hardware and Software Requirements**

These are the Hardware and Software Requirements used to run our application.

**Hardware Requirements**

RAM: 4GB (min)

Hard Disk: 128GB (min)

Keyboard: Standard Windows / MAC Keyboard

Processor: Intel core i5 and above (recommended)

**Software Requirements**

Technologies: Python

IDE: Jupyter Notebook / Google Collab / Visual Studio Code

Operating System: WindowsXP / MacOS

Version: Python 3.11

## **Modules and Libraries Used**

### **NLTK**

Natural Language Toolkit is a Python library for natural language processing (NLP). It provides tools and resources for tasks like tokenization, stemming, tagging, parsing, and classification. NLTK offers a wide range of functionalities for text analysis and linguistic research, making it a popular choice for both beginners and experts in the field. It includes various corpora, lexical resources, and pre-trained models for diverse NLP tasks. NLTK's user-friendly interface and extensive documentation make it easy to use for experimenting with NLP algorithms, developing NLP applications, and teaching NLP concepts. Overall, NLTK is a powerful tool for exploring and analysing textual data in Python.

### **PdfMiner**

It is a Python library for extracting text and metadata from PDF documents. It provides functionalities for parsing PDF files, extracting text content, and accessing document metadata like author, title, and creation date. PDFMiner supports both high-level and low-level access to PDF data, allowing users to extract text with layout preservation or access individual elements like characters, lines, and rectangles. It handles various types of PDF documents, including scanned PDFs, and offers options for customizing text extraction processes. PDFMiner is widely used in fields such as document analysis, information retrieval, and data extraction, making it a valuable tool for working with PDF files programmatically.

### **Textract**

Textract is a Python library that simplifies text extraction from various document formats such as PDFs, images, and more. It provides a unified interface for extracting text from diverse file types, including scanned documents, by leveraging underlying OCR (Optical Character Recognition) engines like Tesseract. Textract abstracts away the complexities of working with different file formats and OCR engines, allowing developers to easily extract text content programmatically. It supports a wide range of document types and provides options for customizing extraction settings. Textract is commonly used in applications requiring automated text extraction from documents, enabling efficient data processing and analysis.

### **io**

The 'io' module in Python provides tools for working with streams of data, including input and output operations. It offers an abstraction layer over various types of input/output sources such as files, sockets, and in-memory buffers. With 'io', developers can read from and write to these sources using a consistent interface, regardless of the underlying implementation. This module facilitates efficient data handling and manipulation, supporting tasks like reading from or writing to files, processing network communication, and managing data streams within memory. Overall, the 'io' module is fundamental for performing input/output operations in a versatile and Pythonic way.

### **ipywidgets**

ipywidgets is a Python library that enables the creation of interactive widgets for Jupyter notebooks and JupyterLab environments. These widgets provide a user-friendly interface for manipulating and visualizing data, enhancing the interactive experience of notebooks. ipywidgets offers a wide range of pre-built widgets such as sliders, buttons, text inputs, and plots, as well as the flexibility to create custom widgets. Users can interact with these widgets directly within the notebook, dynamically updating visualizations and calculations based on user input. ipywidgets is widely used for data exploration, analysis, and visualization tasks, making Jupyter notebooks more engaging and effective for communication and collaboration.

### **NumPy**

NumPy is a fundamental Python library for numerical computing, providing support for multidimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays efficiently. It is the foundation for many other Python libraries in the scientific computing ecosystem. NumPy's key feature is its ndarray (n-dimensional array) object, which allows for fast and efficient array operations. It offers functionalities for array creation, manipulation, indexing, slicing, and broadcasting, making it suitable for a wide range of mathematical and scientific computations. NumPy is extensively used in fields such as data analysis, machine learning, signal processing, and scientific research.

### **Pandas**

Pandas is a powerful Python library widely used for data manipulation and analysis. It provides a fast and flexible data structure called Data Frame, which is a two-dimensional, labelled data structure with columns of potentially different types. Pandas simplifies data handling tasks such as loading, cleaning, transforming, and analysing data from various sources including CSV files, databases, and Excel spreadsheets. It offers a rich set of functions for data exploration, selection, aggregation, and visualization. Pandas also integrates well with other Python libraries like NumPy and Matplotlib, making it a versatile tool for data scientists, analysts, and developers working with structured data.

# **Working Model**

The proposed system implements a basic version of a job recommendation system that works on hybrid filtering model that combines the advantages of two systems i.e., content-based filtering, collaborative filtering. The system can take the input explicitly from the user or else, it can extract the requirements from the pdf or word file that is given as an input to the code. The main features like the job-seeking candidate’s skills, expertise, internships they’ve worked on and experience they have been considered as well as the domain they were interested in.

The code contains several functions that work together to identify the required information from the candidate, either taken manually or by giving their resume as input.

## **Functions Used**

**1. cosineSimilarity (arr1, arr2):** Cosine similarity is a measure used to determine the similarity between two vectors by calculating the cosine of the angle between them. It is commonly used in information retrieval and natural language processing. The formula involves taking the dot product of the vectors and dividing it by the product of their magnitudes. Cosine similarity ranges from -1 to 1, where 1 indicates perfect similarity, 0 indicates no similarity, and -1 indicates complete dissimilarity. It disregards the magnitude of the vectors, focusing solely on their directions. Higher cosine similarity values suggest higher similarity between the vectors, making it a valuable tool for tasks like document similarity and recommendation systems.

**2. jaccardSimilarity (arr1, arr2):** Jaccard similarity is a measure used to compare the similarity between two sets. It is calculated by dividing the size of the intersection of the sets by the size of the union of the sets. In other words, it quantifies the overlap between two sets by considering the proportion of elements they have in common relative to the total number of distinct elements they contain. Jaccard similarity ranges from 0 to 1, where 0 indicates no overlap (complete dissimilarity) and 1 indicates complete overlap (perfect similarity). It is commonly used in various fields, including information retrieval, data mining, and recommendation systems, to compare the similarity between items or documents.

## **Flow of Work**

**Data collection:** Gathering job postings, user profiles, and historical interaction data from various sources, such as job boards and recruitment platforms.

**Preprocessing:** Cleaning and preprocessing the data to ensure consistency and relevance, including text processing and feature extraction.

**Job Matching**: Match user skills and domains with job requirements using predefined rules or criteria. This could include keyword matching or simple filtering based on predefined attributes

**Ranking**: Rank the matched jobs based on relevance to the user's profile. Then storing the Scores in a dictionary for further process.

**Presentation:** Present the matched and ranked job recommendations to the user through a interface such as a console or terminal. Ensure the interface is user-friendly and provides relevant information about each job.

**Deployment:** Integrating the recommendation system into existing job search platforms or building a standalone application for users to access.

**Maintenance and Improvement:** Continuously monitoring and updating the system to adapt to changes in user preferences, job market dynamics, and technological advancements. Overall, the project aims to deliver a robust and scalable job recommendation system that enhances the job search experience for users and improves hiring outcomes for employers by leveraging the power of hybrid filtering models and machine learning algorithms.

## **System Architecture**

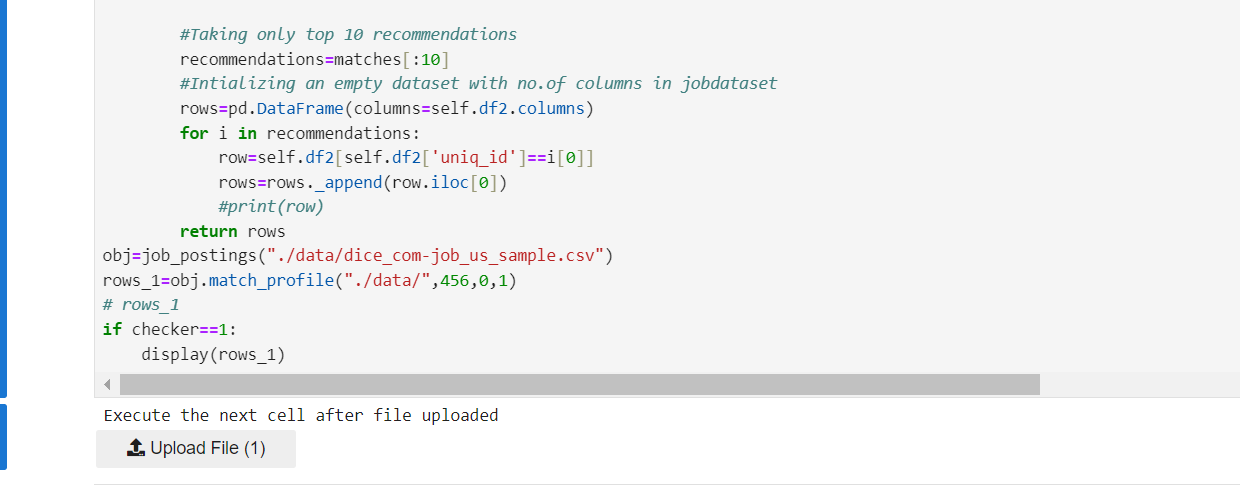
The recommendation system implementation by acquiring a dataset of companies, is a crucial step in the content-based filtering approach. Once the data is gathered, it proceeds to preprocess for analysis. The modules allow the code to quantify the importance of each word in the dataset by considering both its frequency within a specific company profile and its rarity across all profiles. Cosine similarity measures the cosine of the angle between two vectors, in this case, the vectors representing the companies' attributes. This similarity computation results in a similarity matrix, where each entry represents the similarity score between two companies. Utilizing this similarity matrix, the code generates top-N recommendations using a content-based approach. By identifying companies with attributes at a given target company, the system recommends similar companies to users.

Transitioning to Collaborative Filtering, the paper employs two distinct methodologies: a memory-based approach and a model-based approach. In the memory-based approach, the system calculates user-user rating-based similarity. This involves finding the similar user based on skills. For this we use Jaccard similarity functions which takes skills as input and find out the similarity score of target user with all other user. The user with highest score is most similar to target user. Then generate recommendation for the target user based on similar user

After obtaining recommendations from both content-based and collaborative filtering approaches, the program combines the results of these two techniques and give accurate top-N recommendations to users. Ultimately, we get a Hybrid Recommendation System that integrates content-based and collaborative filtering methodologies, offering users a more robust and personalized recommendation experience.

# **Experimental Results**

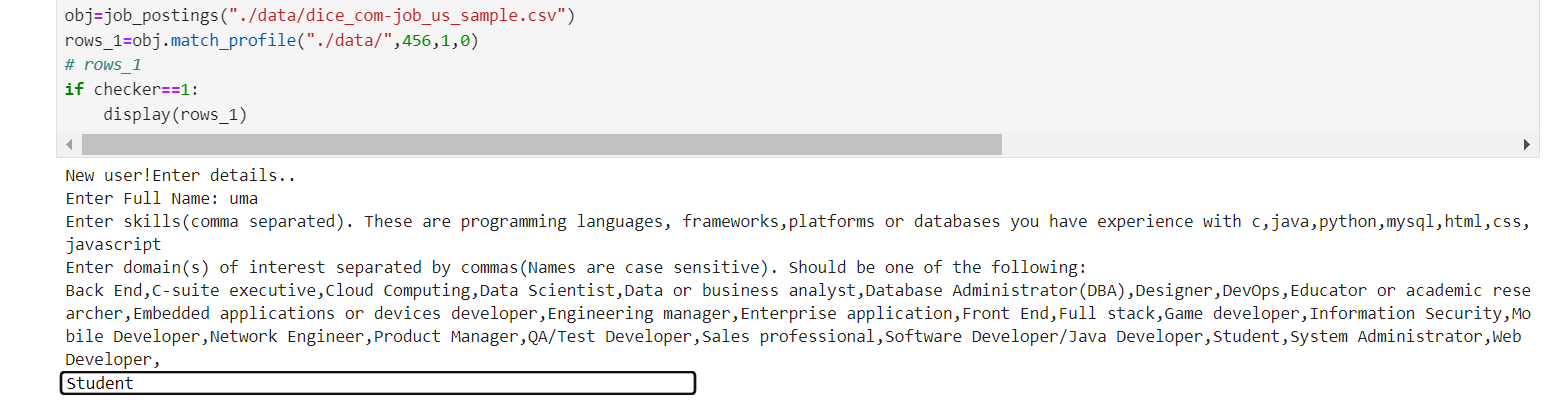
In this section, experiments are performed to verify the effectiveness of the designed Hybrid Filtering Model. Then parameters are discussed. Finally feature comparisons with related models are performed to explain the advantages of this model more efficiently. The proposed method is simulated in Python Jupyter Notebook and the results are obtained as:



**Figure 5.1 To take file as an input**



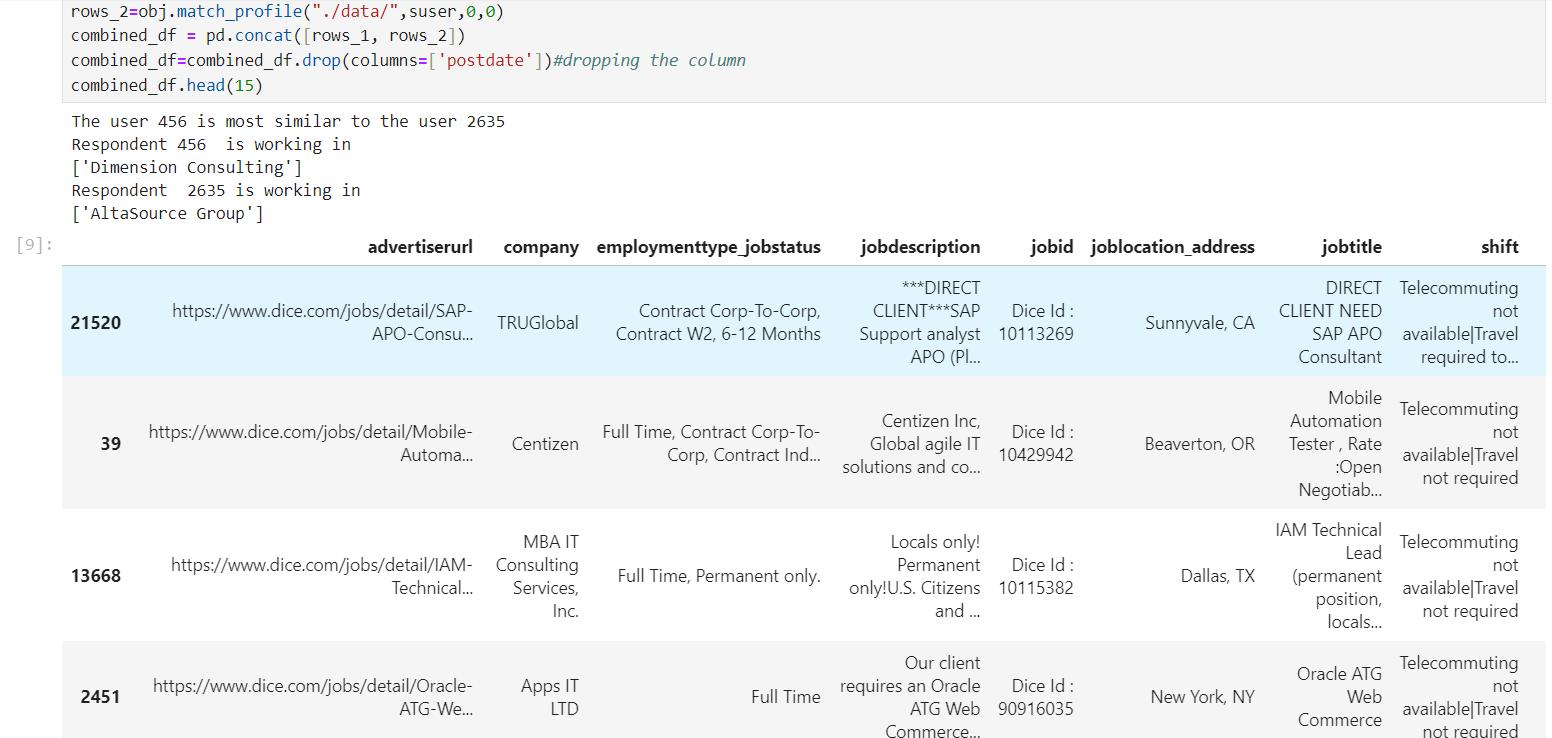
**Figure-5.2 Output after processing resume**



**Figure-5.3 Manual entry of data**



**Figure-5.4 Output after processing user data**

****

**Figure-5.5 Output when input is user\_id**

# **Conclusion and Future Scope**

## **Conclusion**

Hybrid filtering in job recommendation systems combines multiple recommendation techniques to enhance accuracy, coverage, and user experience. By integrating collaborative filtering and content-based filtering, it leverages the strengths of each approach while mitigating their weaknesses. Collaborative filtering analyses user behaviour and preferences to recommend jobs based on similarities between users. However, it often struggles with the cold start problem, where new users or items lack sufficient data for accurate recommendations. Content-based filtering, on the other hand, focuses on the attributes of jobs and users to make recommendations. It can address the cold start issue but may suffer from the sparsity problem when dealing with large datasets.

Overall, hybrid filtering in job recommendation systems offers a powerful solution for matching job seekers with relevant opportunities, benefiting both candidates and employers in navigating the complexities of the job market. The benefits of hybrid filtering include enhanced accuracy, increased coverage of job listings, improved robustness against data sparsity and cold start issues, and better user satisfaction due to personalized recommendations. Furthermore, the modular nature of hybrid filtering makes it scalable and adaptable to evolving data environments and user preferences.

## **Future Scope**

The future scope of hybrid filtering in job recommendation systems is rich with potential for advancement and innovation. One avenue of exploration involves leveraging advanced machine learning techniques, such as deep learning and reinforcement learning, to further refine recommendation accuracy and personalization. These methods can extract intricate patterns and insights from user behaviour and job attributes, enhancing the quality of recommendations. Another promising direction is the integration of contextual information into recommendation algorithms. By considering factors like user location, time of day, and device type, hybrid filtering can deliver more relevant and timely job suggestions, improving user experience and engagement.

Furthermore, the incorporation of multi-modal data types, including text, images, audio, and video, holds promise for providing richer job recommendations. By analyzing diverse sources of information such as job descriptions, resumes, and multimedia content, hybrid filtering can gain a deeper understanding of user preferences and job requirements. Dynamic ensemble methods present another opportunity for improvement, allowing recommendation algorithms to adaptively adjust the weights assigned to different techniques based on their performance and relevance to specific users or scenarios. This approach optimizes recommendation quality and robustness over time.

Lastly, addressing privacy concerns through privacy-preserving recommendation techniques, such as federated learning and differential privacy, will be essential for safeguarding user data while still delivering accurate and personalized recommendations. Overall, these advancements in hybrid filtering have the potential to revolutionize job recommendation systems, empowering both job seekers and employers in navigating the complexities of the job market.

# **7 Bibliography**

|  |  |
| --- | --- |
| [1] | Pradeep Kumar SIngh, Pijush Kanti Dutta Pramanik, Avick Kumar Dey, Prasenjit Choudhury. “Recommender Systems: An Overview, Research Trends, and Future Directions,” 2021. |
| [2] | Dr. Alka Singhal Shivangi Rastogi, Nikhil Panchal, Shivani Chauhan, Shradha Varshney. “Research Paper On Recommendation System,” vol. 9, no. 8, August 2021. |
| [3] | Ravita Midhra, Sheetal Vikram Rathi, “Efficient and Scalable Job Recommender System Using Collaborative Filtering,” *Researchgate,* 19 May 2020. |
| [4] | Tanya V. Yadalam, Vaishnavi M. Govda, Vandhita Shiva Kumar, Disha Girish, “Career Recommendation Systems using Content based Filtering,” in *5th International Conference on Communication and Electronics Systems (ICCES)*, 10 July 202. |
| [5] | Marwa Hussien Mohamed, Mohamed Helmey Khafagy, Mohamed Hasan Ibrahim, “Recommender Systems Challenges and Solutions Survey,” in *International Conference on Innovative Trends in Computer Engineering (ITCE)*, 2 February 2019. |
| [6] | Greg Linden, Brent Smith, Jeremy York, “Amazon.com Recommendation,” in *IEEE Computer Society*, 2003. |
| [7] | Kunal Shah, Akshaykumar Salunke, Saurabh Dongare, Kisandas Antala, “Recommender Systems: An Overview of different approaches to recommendations,” in *International Conference on Innovations in information Embedded and Communication Systems (ICIIECS)*, 2015. |
| [8] | Thiengburanathum P, Cang S, Yu H, “An overview of travel recommendation system,” in *IEEE 22th international conference on automation and computing*, 2016. |
| [9] | J. Karim, “Hybrid Systems for Personalized Recommendations, Research Challenges in Information Science (RICS),” in *2014 IEEE Eighth International Conference*, May 2014. |
| [10] | E K Subramanian, Ramachandran. “Student Career Guidance System: Recommendation of a course,” *International Journal of Recent Technology and Engineering,* vol. 7, no. 6S4, 2019. |
| [11] | Manish Kumar Singh, Dr. Dinesh Prasad Sahu. “Research Aspects Of The System,” *International Journal For Research In Applied Science Engineering Technology (IJRASET),* vol. 5, no. XI, November 2017. |
| [12] | Bhumika Bhatt, Prof. Premal J Patel, Prof. Hetal Gaudani, “A Review Paper on Machine Learning Based Recommendation System,” *International Journal Of Engineering Development And Research,* vol. 2, no. 4, 2014. |
| [13] | Kaveri Roy, Aditi Choudhary, J. Jayapradha, “Product Recommendations using Data Mining And Machine Learning Algorithms,” *ARPN Journal Of Engineering And Applied Sciences ,* vol. 12, no. 19, October 2017. |