A

Mini-Project Report on

Job Recommendation System Using Content-Based Filtering

BACHELOR OF ENGINEERING

IN

Computer Science & Engineering

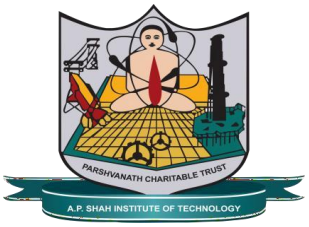
Artificial Intelligence & Machine Learning

By

Mihir Bhave	21106051
Tanisha Chitnis	21106003
Harshal Deshmukh	22206008

Under the guidance of

Prof. Nirali Arora



Department of Computer Science & Engineering

(Artificial Intelligence & Machine Learning)

A. P. Shah Institute of Technology

G.B. Road, Kasarvadavli, Thane (W) - 400615

University of Mumbai

(2024 - 2025)

Abstract

The Smart Job Recommendation System is an intelligent web application designed to simplify the job search process by offering personalized job suggestions based on user input. It employs a content-based filtering approach, utilizing TF-IDF (Term Frequency–Inverse Document Frequency) vectorization and cosine similarity to analyze the relevance between user queries or resume content and thousands of job postings. Users can interact with the system through a user-friendly interface developed using Streamlit, allowing them to input specific keywords or upload resumes in PDF or TXT format. The system extracts textual features from job descriptions and compares them with the user's skills and preferences to generate ranked job recommendations. Additional filters such as location and employment type further refine the results, enhancing accuracy and relevance. This tool serves as a valuable solution for both job seekers and hiring platforms by automating the matching process, improving job visibility, and reducing the time to find suitable employment opportunities.

Keywords: Job Recommendation System, Content-Based Filtering, TF-IDF, Cosine Similarity, Resume Parsing, Job Matching, Natural Language Processing, Streamlit, Career Recommendation, Job Search Automation.

Table of Contents

1. Introduction

In today's fast-paced digital world, job seekers are often overwhelmed by the sheer volume of job postings available across multiple platforms. Manually filtering through these listings to find roles that align with their skills, preferences, and experience is not only time-consuming but also inefficient. At the same time, recruiters face challenges in identifying the most suitable candidates from thousands of applicants.

To address these issues, recommendation systems have emerged as powerful tools across domains such as e-commerce, entertainment, and job search. Among these, Content-Based Recommendation Systems play a crucial role in personalizing user experience by analyzing the textual features of user inputs and matching them with relevant content. In the job market context, such systems can significantly streamline the process by recommending opportunities that align closely with a candidate's resume or search query.

This project, Smart Job Recommendation System, leverages a content-based filtering approach using TF-IDF vectorization and cosine similarity to compare user queries or resume content against job descriptions. Built with Streamlit, the application provides an interactive interface where users can search for jobs by keyword or upload their resumes for automated matching. Additional filters such as location and job type help refine results, ensuring that the recommendations are not just relevant, but also practical.

By combining Natural Language Processing (NLP) techniques with intuitive design, this system aims to enhance the efficiency of the job search process, reduce candidate frustration, and increase the chances of successful employment matching.

2. Literature Survey

The growing reliance on recommendation systems in recruitment platforms has prompted extensive research to enhance the accuracy and contextual relevance of job suggestions. Traditional job portals often rely on keyword-based filtering, which lacks semantic understanding and personalization. In response, more sophisticated techniques involving content-based filtering and natural language processing (NLP) have been widely adopted.

Lops, Gemmis, and Semeraro (2011) discussed the fundamentals of content-based recommendation systems, emphasizing how user profiles can be matched with item descriptions using methods such as Term Frequency-Inverse Document Frequency (TF-IDF) and cosine similarity. These techniques are particularly effective for textual datasets like job postings and resumes, forming the backbone of many modern recommender systems.

Malik, Rauf, and Nayak (2019) introduced a hybrid recommendation system that merges collaborative filtering with content-based techniques, addressing the cold-start problem common in collaborative approaches. However, for applications like resume-job matching, where user interaction data may be sparse or unavailable, content-based methods remain more suitable.

Singh and Saini (2020) explored the application of NLP in parsing resumes and job descriptions to extract relevant features for recommendation. Their work demonstrated the effectiveness of preprocessing and vectorization in improving job-candidate matching performance.

Zhao et al. (2015) explored deep learning-based recommendation models using neural embeddings, which provide better contextual understanding. Despite their high accuracy, such models often demand significant computational resources, making them less viable for real-time web applications and initial project prototypes.

This project builds upon the content-based recommendation paradigm by leveraging TF-IDF and cosine similarity for effective, efficient, and interpretable job matching. The chosen approach balances simplicity with performance, making it ideal for deploying lightweight and interactive applications like a web-based job recommender.

3. Problem Statement

In today's digital age, job portals and professional networking platforms host millions of job postings across various domains, locations, and experience levels. While this abundance of information offers great opportunities, it also presents a significant challenge for job seekers to efficiently navigate through the listings and identify roles that genuinely match their skills, interests, and professional goals.

Most traditional job search engines rely on simple keyword-matching algorithms or basic filtering mechanisms based on location, industry, or employment type. These approaches lack the ability to understand the semantic relevance between a candidate's resume and a job description. As a result, users often receive irrelevant recommendations that lead to frustration, wasted time, and missed opportunities.

Furthermore, many job seekers—especially fresh graduates or those transitioning into new career fields—may not know the exact keywords to search for, making it even harder to find suitable roles using conventional systems. Similarly, recruiters struggle to attract the right candidates, as their job listings may not surface effectively to users who would be a perfect fit but used different terminology in their resumes.

This project addresses the problem by developing a Smart Job Recommendation System that uses Natural Language Processing (NLP) and a content-based filtering approach. The system not only considers keyword input from the user but also accepts resumes in PDF or TXT format and matches them semantically with job descriptions. By analyzing the text data from resumes and job postings, the system identifies the most relevant job opportunities based on context, skills, and experience. It also allows users to apply additional filters such as location and employment type to further refine the recommendations.

The goal is to enhance the job search experience by making it more personalized, efficient, and accurate—bridging the gap between qualified candidates and the right job opportunities through intelligent automation.

4. Proposed System & Implementation

4.1. Model Building & Training

```python
class JobRecommender:
    def __init__(self, csv_path):
        self.df = pd.read_csv(csv_path)
        self.df.dropna(subset=['title', 'description'], inplace=True)
        self.df.reset_index(drop=True, inplace=True)
        self.df['text'] = self.df['title'] + " " + self.df['description'].fillna("") + " " + self.df['skills_desc'].fillna("")
        self.vectorizer = TfidfVectorizer(stop_words='english')
        self.tfidf_matrix = self.vectorizer.fit_transform(self.df['text'])
```

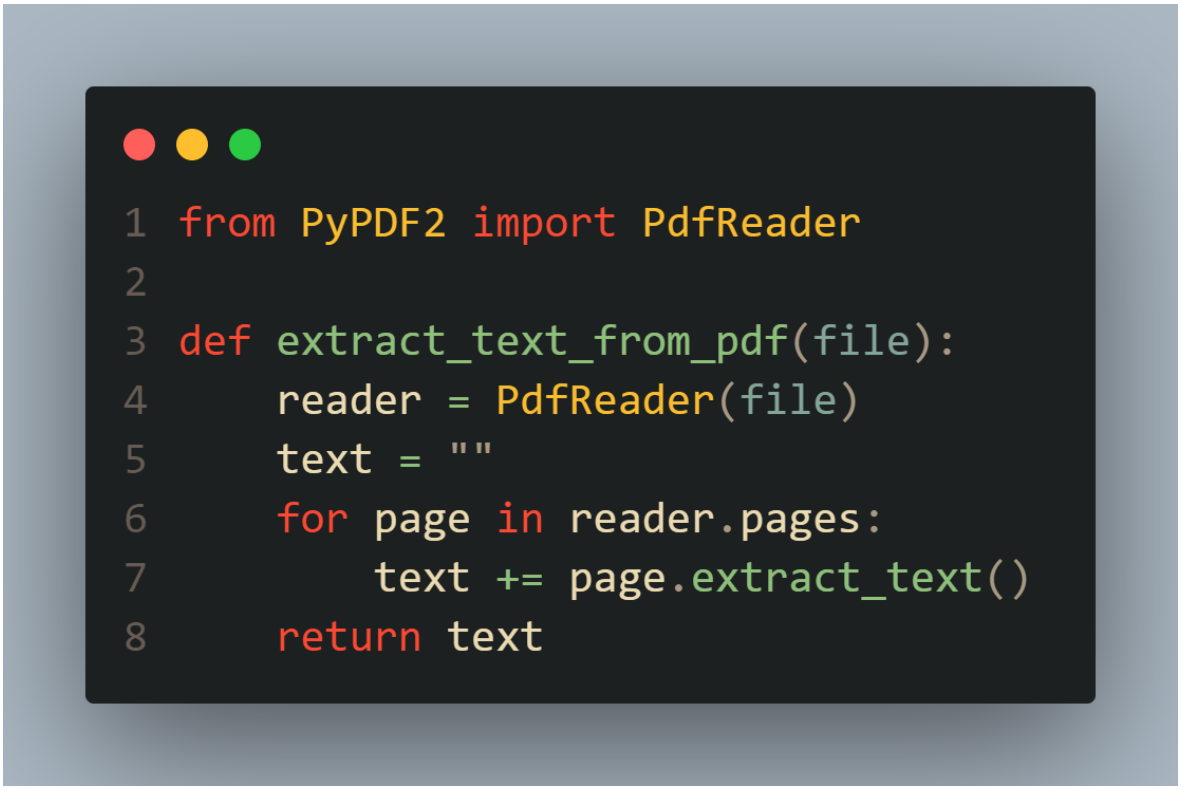
Figure 1: Initializing the TF-IDF Matrix

```python
def recommend(self, user_input, top_n=5, location_filter=None, type_filter=None):
        input_vec = self.vectorizer.transform([user_input])
        cosine_similarities = cosine_similarity(input_vec, self.tfidf_matrix).flatten()
        self.df['similarity_score'] = cosine_similarities

        filtered_df = self.df.copy()
        if location_filter:
            filtered_df = filtered_df[filtered_df['location'].fillna("").str.contains(location_filter, case=False, na=False)]
        if type_filter:
            filtered_df = filtered_df[filtered_df['formatted_work_type'].fillna("").str.contains(type_filter, case=False, na=False)]

        top_indices = filtered_df.sort_values(by='similarity_score', ascending=False).head(top_n).index
        return self.df.loc[top_indices][['title', 'company_name', 'location', 'formatted_work_type', 'description', 'job_posting_url', 'similarity_score']]
```

Figure 2: Generating Recommendation Using the Matrix and User Input



```python
from PyPDF2 import PdfReader

def extract_text_from_pdf(file):
    reader = PdfReader(file)
    text = ""
    for page in reader.pages:
        text += page.extract_text()
    return text
```

Figure 3: Resume Cleaning for Recommendation

4.2. GUI Development

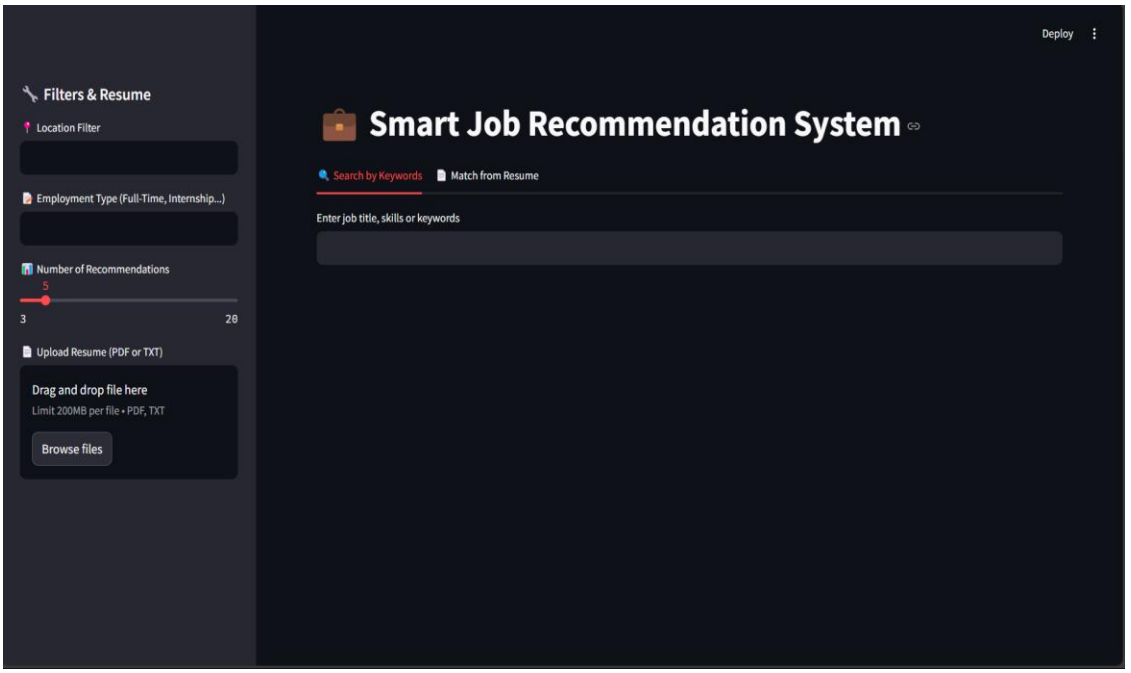


Figure 5: Initial GUI

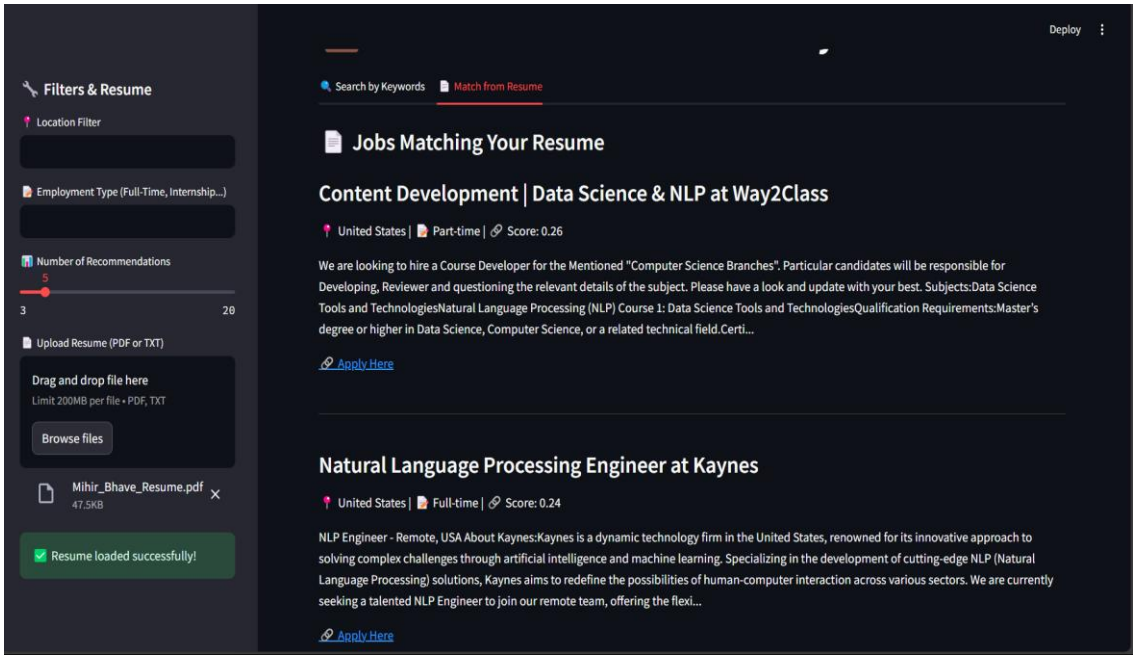


Figure 6: Jobs Based on Resume

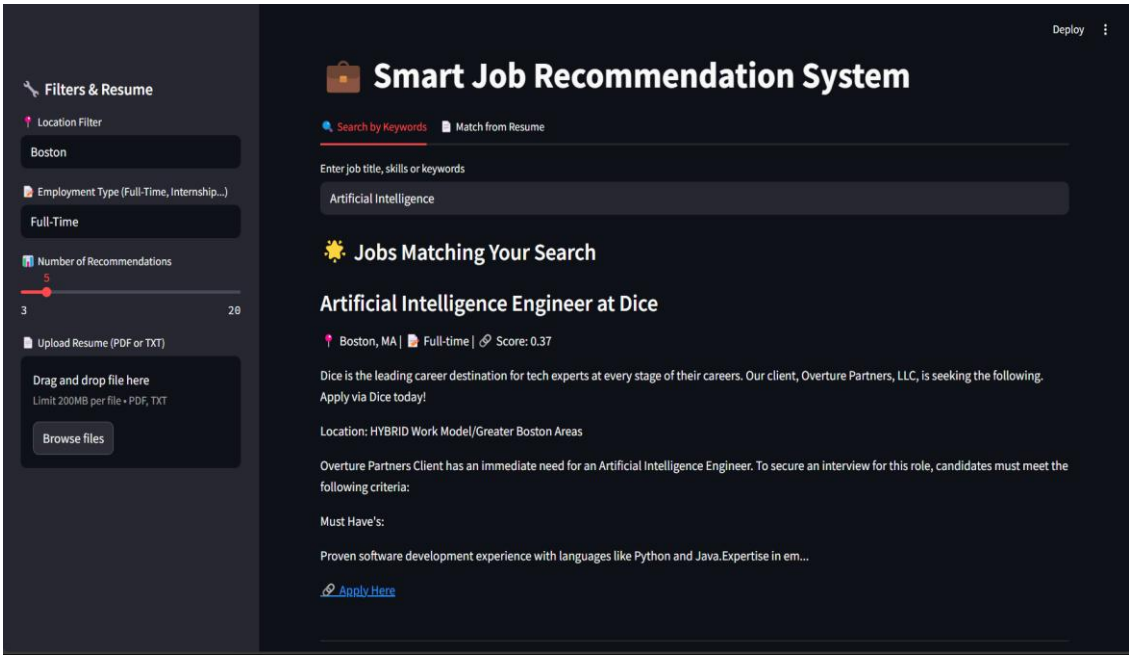


Figure 7: Recommendation Based on Various Filters

5. Conclusion

The Smart Job Recommendation System was developed to tackle a pressing challenge in the modern job market—helping job seekers efficiently discover opportunities that are well-matched to their individual skills, qualifications, and aspirations. Traditional job portals often rely on keyword-based search, leading to mismatches and user frustration. Our system addresses this by adopting a content-based filtering approach powered by Natural Language Processing (NLP), which enables semantic-level matching between job descriptions and user queries or resumes.

The system is built using Streamlit, providing an interactive and responsive user interface that accommodates two main search modes: one based on keyword input and the other based on the analysis of an uploaded resume. This dual-mode approach not only increases flexibility but also reduces dependency on the user's ability to articulate specific roles or skills. Users can fine-tune their searches through filters such as location and employment type, making the experience more personalized and meaningful.

From a technical perspective, the recommender system leverages TF-IDF vectorization to represent textual information in numerical form and cosine similarity to rank job postings based on relevance. This enables precise recommendations without requiring user ratings or historical data, which are often difficult to collect. Additionally, the system processes large volumes of job data efficiently and is designed to be scalable and modular, allowing for easy improvements.

Through this project, we demonstrate that a combination of intuitive UI design, efficient text processing, and intelligent filtering mechanisms can significantly enhance the job search experience. The results are highly relevant job matches presented in a clean and informative layout, reducing the time and effort required to find suitable employment.

In the future, the system can be improved by incorporating collaborative filtering for social recommendations, real-time integration with job listing APIs, feedback-based learning to enhance personalization, and multilingual support to reach broader user bases. The architecture is also adaptable for mobile applications or browser extensions, making it even more accessible.

Overall, this project bridges the gap between technology and career discovery, offering a smarter, faster, and more personalized way for individuals to navigate the ever-growing landscape of job opportunities.

References

[1] Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. ACM Transactions on Computer-Human Interaction (TOCHI), 17(1), 1–46. https://doi.org/10.1145/1100885.1100887

[2] Lops, P., De Gemmis, M., & Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In Recommender Systems Handbook (pp. 73–105). Springer. https://doi.org/10.1007/978-0-387-85820-3_3

[3] Malinowski, J., Keim, T., Wendt, O., & Weitzel, T. (2006). Matching people and jobs: A bilateral recommendation approach. Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06), 137b–137b. https://doi.org/10.1109/HICSS.2006.196

[4] Paparrizos, J., Cambazoglu, B. B., & Gionis, A. (2011). Machine learned job recommendation. Proceedings of the Fifth ACM Conference on Recommender Systems (RecSys), 325–328. https://doi.org/10.1145/2043932.2043991

[5] Ziegler, C. N., McNee, S. M., Konstan, J. A., & Lausen, G. (2005). Improving recommendation lists through topic diversification. Proceedings of the 14th International Conference on World Wide Web (WWW), 22–32. https://doi.org/10.1145/1060745.1060754

[6] Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. Information Processing & Management, 24(5), 513–523. https://doi.org/10.1016/0306-4573(88)90021-0

[7] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

[8] Zhang, Y., & Chen, X. (2020). Explainable recommendation: A survey and new perspectives. Foundations and Trends® in Information Retrieval, 14(1), 1–101. https://doi.org/10.1561/1500000066

[9] Schafer, J. B., Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative filtering recommender systems. In The Adaptive Web (pp. 291–324). Springer. https://doi.org/10.1007/978-3-540-72079-9_9

[10] Beel, J., Gipp, B., Langer, S., & Breitinger, C. (2016). Research-paper recommender systems: A literature survey. International Journal on Digital Libraries, 17(4), 305–338. https://doi.org/10.1007/s00799-015-0156-0

[11] Ghazanfar, M. A., & Prugel-Bennett, A. (2010). An improved switching hybrid recommender system using Naive Bayes classifier and collaborative filtering. Proceedings of the 2010 ACM Conference on Recommender Systems (RecSys), 257–260. https://doi.org/10.1145/1864708.1864763

[12] Aggarwal, C. C. (2016). Recommender systems: The textbook. Springer. https://doi.org/10.1007/978-3-319-29659-3

[13] Liu, Y., Dolan, P., & Pedersen, E. R. (2010). Personalized news recommendation based on click behavior. Proceedings of the 15th International Conference on Intelligent User Interfaces, 31–40. https://doi.org/10.1145/1719970.1719976

[14] Rafter, R., Bradley, K., & Smyth, B. (2000). Personalized retrieval for online recruitment services. Proceedings of the International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, 24–34. https://doi.org/10.1007/3-540-44595-1_3

[15] Hu, Y., Koren, Y., & Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. 2008 Eighth IEEE International Conference on Data Mining, 263–272. https://doi.org/10.1109/ICDM.2008.22