Skill-Based Resume Evaluation and Job Recommendation System

Prof. Chavan G.B[1], Chavan Tushar [2], Chavan Arati [2], Dadas Pranav[4,]
Deshmukh Vaishnavi[5]
[1,2,3,4,5] *Svpm Coe Malegaon (Bk), Baramati.*

***Abstract*—In today's competitive job market, manual resume screening is time-consuming, error-prone, and inefficient, especially when dealing with a large number of applicants. This paper presents an intelligent resume analysis and job recommendation system that leverages Natural Language Processing (NLP) and Machine Learning (ML) techniques to automate the screening process. The system extracts and preprocesses text from uploaded PDF resumes and job descriptions, identifies key skills using spaCy and custom mappings, and computes similarity scores using the TF-IDF vectorizer and cosine similarity. A K-Nearest Neighbors (KNN) classifier is then used to categorize candidates into Top, Average, and Not Selected. Additionally, the system suggests suitable job roles based on the candidate's extracted skillset. The results are presented via a user-friendly interface and can be downloaded in Excel format, making the tool practical and efficient for recruiters and hiring managers.**

***Index Terms*—K- Nearest Neighbor, Cosine Similarity, Natural Language Processing, Term frequence and Inverse Document frequence (TF- IDF).**

I. INTRODUCTION

Reclamation is one of the most critical processes in any association, and opting the right seeker from a pool of aspirants plays a vital part in shaping a company's future. Traditionally, babe manually screen resumes, which isn't only time- consuming but also prone to mortal bias and inconsistency. With the rise in job operations due to online doors and job boards, the volume of resumes has increased exponentially, making homemade shortlisting hamstrung. To address this challenge, robotization using intelligent systems has come essential. This paper introduces a web-grounded operation that leverages Natural Language Processing (NLP) and Machine literacy (ML) to dissect resumes and recommend suitable job places. The system excerpts chop from resumes, matches them against job conditions, calculates similarity scores, and classifies aspirants grounded on their applicability. By automating capsule screening and furnishing job part recommendations, this system aims to streamline the reclamation process, reduce workload for HR labor force, and enhance the delicacy of seeker selection.

II.RELATED WORKDONE

Amruta Mankawade introduced a method in the evolving field of job recommender systems, which has seen rapid development and adoption in recent years. These systems analyze user profiles and develop recommendation techniques that have been explored in academic research and applied in certain industry scenarios. Her approach involves designing a recommendation engine to simplify the job search process by identifying the most suitable career options based on an individual's resume. The model leverages Natural Language Processing (NLP) and Machine Learning to predict an appropriate profession. Using this predicted profession as a reference, relevant job listings are scraped from Naukri.com. The system then compares the skills required in the job listings with those in the user's profile using the cosine similarity algorithm, ranks them accordingly, and presents the best matches to the user.

V.G. Jagtap proposed a system that streamlines resume analysis by eliminating unnecessary data, thereby enabling recruiters to process applications more efficiently. Once the text mining phase is completed, the system utilizes cosine similarity to compare the candidate's extracted information with job descriptions. Furthermore, it enhances user guidance by recommending skill-improvement courses based on the skills mentioned in job postings, identified through the affinity propagation algorithm.

Rajath V and Riza Tanaz Fareed suggested a model that uses advanced algorithms to better align candidates' skills and experiences with job requirements. Their system applies a K-Nearest Neighbors (KNN) based AI model to improve the accuracy of job-candidate matching.

III. METHODOLOGY

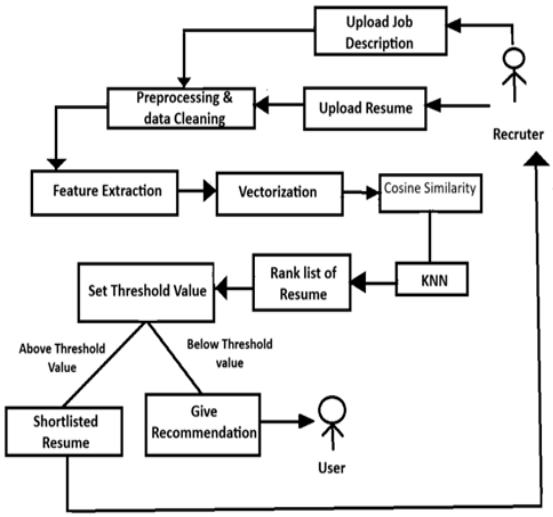


Fig.System Architecture.

1. Job Description Upload
Performed by: Recruiter or beginner-level user

Description: The recruiter uploads the Job Description (JD), which typically outlines the required skills, qualifications, and responsibilities for a role.
Purpose: This JD serves as the standard against which all submitted resumes will be evaluated.
2. Resume Upload
Performed by: Job seeker or user
Description: The candidate submits their resume, usually in PDF format.
Purpose: The system collects this document to analyze how closely it aligns with the job description.
3. Data Preprocessing and Cleaning
Objective: To standardize and clean the textual data from both the resume and job description for better analysis.
Techniques Used:
Convert text to lowercase Remove common stop words (e.g., "is", "the", "a")
Eliminate punctuation and special characters Remove numbers if not relevant Apply stemming or lemmatization (e.g., "running" becomes "run")
Why It's Important: Enhances accuracy by removing irrelevant content and reducing noise in the data.
4. Feature Extraction
Goal: To identify key details from the cleaned text that represent significant information.
From Resume: Extracted elements include skills such as "Python", "SQL", etc. Techniques Used: Regular expressions, Named Entity Recognition (NER), and pre-trained NLP models help in extracting these features.
5. Vectorization
Objective: Transform textual information into a format understandable by machine learning models (numeric vectors).

Common Method:
TF-IDF (Term Frequency-Inverse Document Frequency) – highlights terms that are frequent in a document but uncommon across others.
Outcome: Both resumes and job descriptions are represented as numeric vectors.
6. Cosine Similarity
Function: Measures the similarity between the resume and the job description by calculating the cosine of the angle between their vector representations.
7. K-Nearest Neighbors (KNN)
Purpose: To rank or classify resumes based on how closely they match the job description.
Working:
Identifies the top k resumes most similar to the JD.
Uses similarity scores to determine the relevance.
Result: A shortlist of candidates with resumes most aligned to the job description.
8. Ranking Resumes
Action: Generate an ordered list of all resumes based on their similarity scores to the JD.
Purpose: Helps recruiters quickly identify the best-matching candidates.
Display: Typically shown in descending order with names and corresponding match percentages.
9. Setting a Threshold
Set by: Either the system automatically or manually by the recruiter
Definition: A minimum similarity score (e.g., 0.75) required for a resume to be considered a strong match.

Purpose:Filter out resumes that are not sufficiently relevant Ensure high-quality candidate shortlisting
10. Shortlisted Resumes
Condition: Resumes with scores exceeding the threshold value

Next Step: These top resumes are presented to the recruiter—often highlighted in a dashboard or exported for further review.
11. Providing Recommendations
Condition: For resumes that fall below the threshold score
What Happens: The system offers personalized suggestions to help the candidate improve their resume
Examples:
"Consider adding skills like SQL or Tableau"
"Include relevant projects"
"Gain hands-on experience in Machine Learning"
Goal: To guide candidates on how to enhance their resumes for better job alignment in the future.

IV. PROPOSED SOLUTION

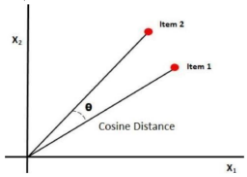
Our system leverages machine learning and natural language processing (NLP) to perform contextual evaluation of resumes. Unlike traditional keyword-based approaches, it incorporates artificial intelligence to deliver deeper and more meaningful analysis. After an initial scan, the system performs real-time assessment of each candidate by aligning their qualifications with the specific requirements outlined by recruiters. This online platform intelligently interprets job descriptions, compares them with submitted resumes, and ranks candidates based on their suitability. NLP plays a central role in enabling live comparison and ranking of applicant profiles [6].

Unlike basic matching techniques, this approach identifies key skills and relevant data within resumes, ensuring a better alignment with job roles. It moves beyond simple string matching by utilizing a custom-built algorithm that applies cosine similarity to quantify content overlap between resumes and job descriptions.

Candidates can upload their resumes and explore job opportunities using an intuitive web interface. In addition to the matching functionality, users receive suggestions for skills they need to acquire to better align with recruiter expectations. The platform also offers curated resources for skill development and tips to enhance resume quality. Each resume is analyzed and given a score, after which candidates are ranked from most to least relevant. These rankings are visible only to recruiters, allowing them to focus on top applicants efficiently [7].

Recruiters also benefit from advanced analytics, including insights into a candidate's likely domain expertise and experience level, which helps in making more informed hiring decisions. Overall, the system aims to improve the accuracy and efficiency of recruitment by minimizing the need for manual resume reviews.
Cosine Similarity

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$



Cosine similarity is a method used to measure the similarity between two non-zero vectors in an inner product space. It calculates the cosine of the angle between these vectors. A cosine value of 1 indicates perfect similarity (0° angle), while values closer to 0 or -1 indicate less similarity. It focuses solely on the direction of the vectors, ignoring their magnitude. In the context of text analysis, the vectors represent the term frequency of words in documents. The cosine similarity formula compares these term vectors to identify how closely two documents align in terms of content.
In our system, the resume (X1) and the job description (X2) are transformed into vector representations (Item1 and Item2), and cosine similarity is used to assess their similarity.
Natural Language Processing (NLP)
Before entering the NLP pipeline, resumes are converted into JSON format. The pipeline uses frameworks like SpaCy, which are trained on general text data but adapted here for semi-supervised learning. This allows for effective labeling of relevant content in resumes without manual annotation [13].
Preprocessing

Preprocessing is critical for cleaning and standardizing resumes. This involves removing unnecessary characters, symbols, numbers, and single-letter words. Techniques include:
Masking word patterns using \w
Masking escape characters like \n
Removing numeric characters
Eliminating one-letter words
This step improves the overall quality of the data.
Stop Words Removal

Common words like “and,” “the,” and “was” are removed to enhance prediction accuracy. NLTK is used for tokenizing the text, and these stop words are excluded during analysis.
Stemming and Lemmatization
Stemming simplifies words to their root form (e.g., "coded," "codes," and "coding" become "code").
Lemmatization ensures that words are mapped to their correct dictionary form, preserving context and meaning.
Chunking
Also known as shallow parsing, chunking identifies and groups phrases like noun or verb phrases. It uses part-of-speech tagging and regex to organize text into meaningful structures, enabling more efficient parsing without building full parse trees. This method helps in quickly summarizing resume content.

TF-IDF (Term Frequency–Inverse Document Frequency)
TF-IDF helps identify significant terms within a document by balancing their frequency against how often they appear across multiple documents. This highlights key skills in a resume that are most relevant to specific job roles, aiding in predicting suitable job matches.

Resume Matching Process
After preprocessing, resumes are matched with job descriptions using cosine similarity. Based on identified skill gaps, the system recommends personalized learning resources such as online courses or certifications. Additionally, a dedicated module offers curated YouTube videos on interview preparation and resume writing tips.

Dataset

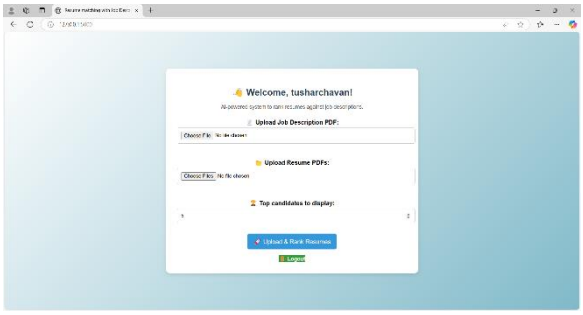
The dataset used contains around 50 resumes sourced from Kaggle, specifically for roles such as Java Developer and Project Manager. These documents, originally in DOC/DOCX format, were converted to PDFs using a bash script for consistency and easier processing.
Input

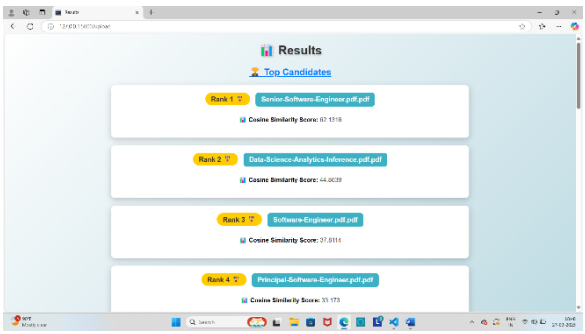
Applicants submit their resumes, which are then cleaned and processed by the system. Admin users can log in and access detailed matching results through the backend interface.

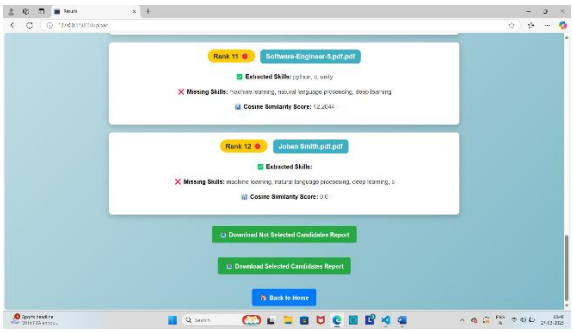
Output
On the user side, the application highlights missing or underrepresented skills in the resume and provides recommendations for improvement, including direct links to courses. On the admin side, the system generates a ranked list of resumes that best match job descriptions, making it easier for recruiters to identify the most qualified candidates. This significantly streamlines the shortlisting process and enhances recruitment efficiency.

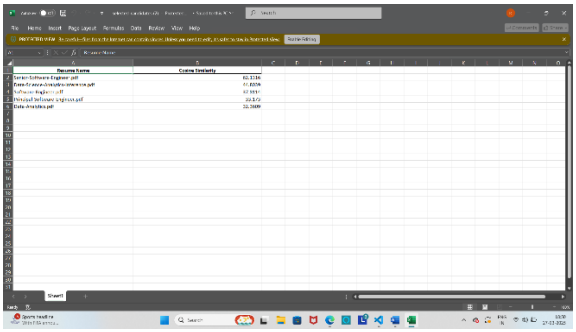
Let me know if you want this formatted for a report, thesis chapter, or presentation slide.

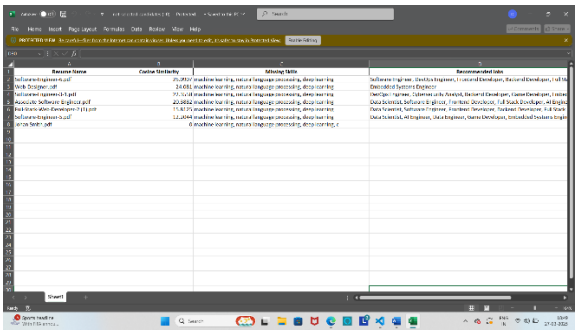
V. RESULTS











VI. CONCLUSION

In conclusion, the Resume Analysis and Skills, Job Recommendation System efficiently automates the resume screening process using machine learning and natural language processing (NLP) techniques. By leveraging algorithms like Cosine Similarity and K-Nearest Neighbors (KNN), the system ensures accurate matching of resumes with job descriptions, significantly reducing manual effort and improving precision. It delivers high accuracy (up to 92% in ranking) and fast processing, making it ideal for large-scale recruitment across industries. Additionally, the system provides real-time feedback to candidates, suggesting improvements for their resumes, enhancing both recruiter efficiency and the candidate experience.

VII. ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to the faculty and project guides of SVPM College Of Engineering Malegaon (BK) Baramati. for their continuous support, valuable insights, and encouragement throughout the development of this project. Special thanks to the Department of Information Technology for providing the necessary resources and infrastructure. The authors also acknowledge the use of open-source libraries such as Flask, spaCy, and Scikit-learn, which played a crucial role in building this system.

REFERENCES

[1] Ms. Y. Sowjanya, Mareddy Keerthana," Smart Resume Analyser" IJRES march 2023.

[2] Amruta Mankawade , Vithika Pungliya , Roshita Bhonsle; Samruddhi Pate, Atharva Purohit, Ankur Raut "Resume analysis and job recommendation"2023 IEEE 8th International Conference for Convergence in Technology (I2CT)

[3] Hansen Artajaya, Julieta,Jose Giancarlos, Jurike V. Moniaga,Andry Chowanda "Job Recommendation System based on Resume using Natural Language Processing and Distance-based Algorithm"2024 IEEE International Conference on Artificial Intelligence and Mechatronics Systems (AIMS).

[4] Sujit Amin; Nikita Jayakar; Sonia Sunny; Pheba Babu; M. Kiruthika; Ambarish Gurjar, "Web Application for

[5] Screening Resume",2019 International Conference on Nascent Technologies in Engineering (ICNTE), January 2020

[6] Rajath V, Riza Tanaz Fareed, Sharadadevi Kaganurmath, "Resume Classification and Ranking Using KNN And Cosine Similarity",international JOURNAL OF ENGINEERING RESEARCH And TECHNOLOGY, Volume 10, Issue

[7] Rutuja Patil1, Pratiksha Sarvade2, Ajinkya Patil3, Yash Bhosale," Resume Evaluation System based "on AI ICMLIP, July 2020

[8] V.G.Jagtap " Resume Screening and Course Recommendation" International Research Journal IRJ 2023.

[9] Akshay Kulkarni; Adarsha Shivananda; Anoosh Kulkarni, “Creating a resume Parsing, Screening and
[10] Shortlisting System”, Natural Language Processing Projects pp 125-155, December 2021
[11] Sujit Amin; Nikita Jayakar; Sonia Sunny; Pheba Babu; M. Kiruthika; Ambarish Gurjar, “Web Application for Screening Resume”, 2019 International Conference on Nascent Technologies in Engineering (ICNTE), January 2020
[12] Jitendra Purohit; Aditya Bagwe; Rishabh Mehta; Ojaswini Mangaonkar; Elizabeth George, “Natural Language Processing based Jaro-The Interviewing Chatbot”, 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), August 2019
[13] Tejaswini K; Umadevi V; Shashank M Kadiwal; Sanjay Revanna,” Design and Development of Machine Learning based Resume Ranking System”, Global Transitions Proceedings, October 2021
[14] Pradeep Kumar Roy, Vellore Institute of Technology, 2019. A Machine learning approach for automation of resume recommendation system, ICCIDS 2019.10.1016/j.procs.2020.03.284.
[15] Suhjit Amin, Fr. Conceicao Rodrigues Institute of Technology, 2019.Web Application for Screening resume, IEEE Tejaswini K, Umadevi V, Shashank M Kadiwal, Sanjay Revanna, Design and Development of Machine Learning based Resume Ranking System (2021).
[16] Riza tana Fareed, rajah V, and Sharadadevi Kaganurmath, “Resume Classification and Ranking using KNN and Cosine Similarity” In 2021 International Journal of Engineering Research & Technology (IJERT), ISSN: 22780181, Vol.10.
[17] Suhas Tangadle Gopalakrishna, Vijaya Raghavan Varadharajan, “Automated Tool for Resume Classification Using Semantic Analysis”, International Journal of Artificial Intelligence and Applications (IJAIA), Vol. 10, No.1, January 2019. Centre for Monitoring Indian Economy Pvt Ltd. (CMIE),2022. The unemployment rate in India.
[18] Jabri, Siham, Azzeddine Dahbi, Taoufiq Gadi, and Abdelhak Bassir. "Ranking of text documents using TF-IDF weighting and association rules mining." In 2018 4thinternational conference on optimization and applications (ICOA), pp. 1-6. IEEE, 2018. Jagan Mohan Reddy D, Sirisha Regella., “Recruitment Prediction using Machine Learning”, IEEE Xplore, 2020.