iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** II **Month of publication:** February 2024

DOI: https://doi.org/10.22214/ijraset.2024.58505

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue II Feb 2024- Available at www.ijraset.com

Advancing Job Search: A Comprehensive Resume-Based Job Recommendation System Using NLP and Deep Learning Techniques

Anish Kulkarni[1], Om Kenge[2], Rohit Mehatre[3], Rucha Kulkarni[4]
MMCOE

Abstract: ***Students find it difficult to find jobs and postings related to their skills and which has a good pay. Students aimlessly scan all internet to find a job which best matches their skill. Hence this paper presents a model which would give them personalized recommendations based on their skills mentioned in the resume. In today's world hundreds of people apply for a single job posting. Companies receive thousands of applications at any given time. To save time companies, recruiters have few seconds to go through the applicant's resume. Moreover, today all companies use a software based application which scans all resumes and get the best candidates. Most students especially freshers fail at this critical juncture, they fail to make a good resume which would help their resumes getting selected. Our novel model helps solve this problem. In order to extract section-specific text content from resumes, this work presents a layout-aware resume parsing system based on natural language processing (NLP) and rule-based approaches. This output can be fed into a resume content review system to obtain resume feedback, or it can be utilized as the input for a resume content score model. By utilizing sophisticated natural language processing (NLP) models, the system guarantees precise recognition and classification of essential resume elements, including personal information, qualifications, experience, and accomplishments. The recommendation engine makes use of machine learning techniques to pinpoint areas that need work, such improving keyword relevancy, recommending extra abilities, or offering formatting and wording advice. The goal of this iterative process is to provide job seekers with dynamic tools that will boost their visibility to recruiters and applicant tracking systems. Our unique model also recommends skills to add in their resume. This models also recommends some add-on courses to improve those skills. The app also uses cutting-edge technology like Natural Language Processing (NLP) and Optical Character Recognition (OCR) to process resumes and job listings and find the greatest fit for both job seekers.***

Keywords: ***Job Recommendation System, Natural Language Processing, Resume Parser, Personalized Suggestions, Word Embeddings***

I. INTRODUCTION

In the present digital era, resume selection is critical to a successful hiring process. Employers who advertise job openings get inundated with resumes. Manually reviewing these resumes takes a lot of time and resources.The conventional method of creating resumes has long been a laborious, subjective, and manual procedure that presents issues with consistency, relevancy, and adaptability. Job seekers face challenges when it comes to customizing their resumes to meet the requirements of various industries and employers. This can result in lost opportunities and frustrations when navigating the competitive job market. Students who are still in college don't necessarily know how to build a resume which would get them selected. Fundamentally, this model makes use of state-of-the-art natural language processing (NLP) methods to effectively extract and comprehend data from unstructured resumes. This feature allows the system to correctly parse a variety of document formats, guaranteeing a seamless and standardization of the extraction of important information, including contact information, educational background, employment history, abilities, and achievements. The Importance of Deep Learning and NLP: Techniques like natural language processing (NLP) and deep learning have changed the game when it comes to hiring and developing talent. With the use of these tools, resumes and job postings can be analyzed more thoroughly and contextually. Students would get an accurate analyses of the score of their resumes. The student must first upload their resumes in PDF format and parse it . The model will then remove stopwords and get a list of keywords. The model will scan it and and score it based on the skills mentioned in the resume .These guidelines cover a wide range of topics, from skill enhancement and keyword relevancy to formatting quirks, guaranteeing that resumes are not just properly formatted but also in line with the ever-changing demands of contemporary hiring systems.

Thus, the purpose of this project is to develop an automated system that uses multiple machine learning algorithms and natural language processing techniques to comprehend, analyze, and make decisions based on the content of resumes. Our Resume Matching Framework's Goals: Our suggested Resume Matching Framework's main objective is to greatly increase the effectiveness and efficiency of the resume-to-job matching procedure. The following goals are addressed by this framework by utilizing the capabilities of NLP and Deep Learning: Semantic analysis: To extract the semantic meaning from job descriptions and resumes, enabling a deeper comprehension of the skills and needs for each position. Customization: To enable the framework to be adjusted to different business sectors and organizational requirements while guaranteeing that it complies with particular recruiting standards. Efficiency: To minimize the time and effort needed for both businesses and job searchers by streamlining and expediting the hiring process.

II. WORKING

A. *OCR*

In machine vision, which includes image processing, artificial intelligence, and pattern identification, optical character recognition (OCR) is essential. OCR has attracted increased attention in light of the popularity of smartphones, especially when it comes to text recognition in natural environments. In the past, OCR relied on features that were manually created, which produced inconsistent results. But OCR was transformed by deep learning, which made it possible to automatically extract features from enormous datasets. Text identification presents difficulties, particularly in natural situations, and frequently calls for modifications to pre-existing networks such as VGGNet or ResNet. Segmenting and categorizing individual characters was the traditional method for text recognition, but it was prone to accumulated errors. Deep learning has taken the lead in OCR today, providing end-to-end solutions without the need for explicit text segmentation. End-to-end networks outperform more conventional techniques in terms of accuracy and robustness by translating text into sequences.

B. *NLP*

Natural Language Processing (NLP) encompasses various techniques, including word tokenization, stop words removal, lemmatization, and bigram collection finder, aimed at understanding, interpreting, and generating human language. Word tokenization involves breaking down text into smaller units or tokens for analysis. Stop words removal eliminates common words like "the" or "and" to focus on meaningful content. Lemmatization reduces words to their base or root form to standardize vocabulary. Bigram collection finder identifies pairs of consecutive words, aiding in understanding contextual relationships. These techniques collectively empower NLP systems to process and extract meaningful information from text data, facilitating applications like chatbots, sentiment analysis, and language translation. Leveraging machine learning, deep learning, and statistical modeling, NLP plays a crucial role in various domains such as healthcare and finance. This concise overview underscores the foundational role of NLP techniques in language processing research and applications.

1) *Text Parsing:* One of the most important steps in shortlisting resumes is text parsing from PDF files to strings. Text can be extracted from PDF files, which are frequently used for documents like job descriptions and resumes. This enables for additional processing and analysis. A well-liked Python module for parsing PDF files and extracting text from them is the pdfplumber library.
2) *Stop word Removal:* Usually, stop word removal comes next in the text processing pipeline after the PDF file has been converted to a text string. Stop words are words like "the," "a," and "and" that are commonly used in the English language but usually don't add much to the text's content. Eliminating these stop words can increase the accuracy of natural language processing models and help the text become less noisy. Many Python packages provide stop-word removal capability. One such library that provides a list of stop words for numerous languages, including English, is the Natural Language Toolkit (NLTK).
3) *Lemmatization:* Lemmatization is the next stage of preparing the text after the stop words have been eliminated. Reducing words to their lemma—their basic or root form—is known as lemmatization. As a result, the dimensionality of the data is decreased by combining similar terms. The Natural Language Toolkit (NLTK) in Python can be used for lemmatization. Tokenizing the words with the NLTK tokenizer is the first step. Next, a part-of-speech (POS) tagger is used to determine the part of speech for each word. This is important because the lemmatizer has to know the part of speech in order to recognize the lemma. The words can be lemmatized using the NLTK lemmatizer after the POS tags have been found. The term and the part of speech are the two arguments that the lemmatizer accepts. When a part of speech is missing, the lemmatizer considers the word to be a noun. Eliminating any remaining punctuation and numbers is essential after the words have been lemmatized. Regular expressions in Python can be utilized for this purpose.

Following the removal of stop words, the lemmatization process often goes like this:

a) Use the NLTK tokenizer to tokenize the words.
b) To ascertain each word's parts of speech, use a POS tagger.
c) Lemmatize the words with the NLTK lemmatizer, making sure to indicate each word's part of speech.
d) Use regular expressions to get rid of any residual punctuation and numeric characters.

Lemmatization is performed to clean up and improve the reliability of the text data once stop words are removed.

C. Working of Recommendation System

1) TF-IDF

For natural language processing applications like text classification, text clustering, and information retrieval, TF-IDF is a popular approach. It is used to extract pertinent keywords and phrases from job descriptions and resumes in the context of resume shortlisting in order to calculate their similarity scores.Making a document-term matrix that shows the frequency of each term in each document is the first step in using TF-IDF. Every term in job descriptions and resumes is a lemma that was discovered using the lemmatization process.One way to see the document-term matrix is as a table where each document is represented by a row and each term by a column.

The next step is to use the TF-IDF formula to calculate the weight of each term in each document after the document-term matrix has been created. The TF-IDF is composed of inverse document frequency (IDF) and term frequency (TF). The inverse document frequency evaluates how much information a word contributes across all documents, whereas the term frequency measures how frequently a phrase appears in a document. The following formula can be used to calculate

TF-IDF:

TF-IDF(term, document) = TF(term, document) * IDF(term)

Were,

TF (term, document) is the frequency of the term in the document.

IDF (term) is the inverse document frequency of the term across all documents.

2) Cosine Similarity

One metric that is used to compare two vectors in a high-dimensional space is cosine similarity. In natural language processing and information retrieval, the method of comparing documents based on their vector representation is commonly employed to determine their similarity.Cosine similarity is a tool used in resume shortlisting processes to assess how close a candidate's resume is to the job description.

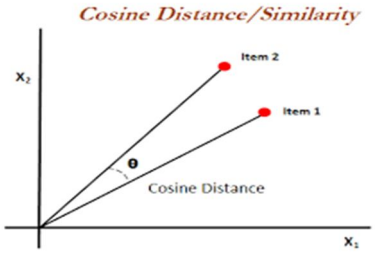
TF-IDF is used to transform the text data into a matrix format once text preprocessing steps including tokenization, lemmatization, and stop word removal are finished. Each term's frequency of occurrence in the text data is included in the matrix, together with a weight assigned to it based on its importance across the file and the corpus as a whole.The similarity between the candidate's resume and the job description is found using cosine similarity once the matrix has been obtained. Between 0 and 1, which represent total similarity and complete similarity between the two vectors, lies the cosine similarity value.

To find the cosine similarity, compute the dot products of the two vectors and divide their magnitude products. To show how similar the two vectors are to each other, the cosine of the angle produced by them is given. Alternatively, the cosine of the angle that the two vectors create in vector space is represented by the cosine similarity value.

One may find the cosine similarity in Python by using the scikit-learn module. Using the TF-IDF method, the text data is transformed into a matrix format using the Scikit-Learn "TfidfVectorizer" function. The "cosine_similarity" function from the same library is used to calculate the cosine similarity between the two vectors.

The following are the steps needed to calculate cosine similarity:

1) Use the "TfidfVectorizer" function to turn the candidate's résumé and the job description into vector representations.
2) To find the cosine terms of similarity between the two vectors, use the "cosine_similarity" function.
3) The cosine similarity value, which is between 0 and 1, should be obtained.
4) A greater degree of resemblance between the candidate's résumé and the job description is indicated by a higher cosine similarity value.



5) KNN Classifier

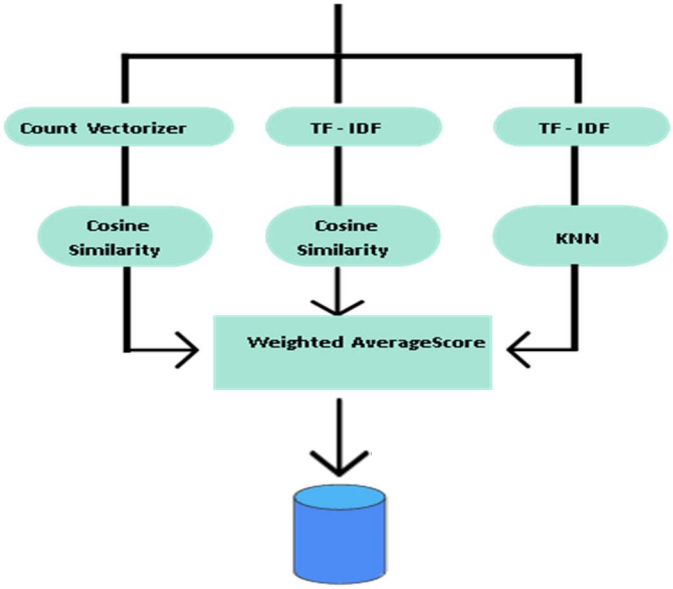
The K-Nearest Neighbors (KNN) algorithm is a versatile and intuitive supervised learning method used for both classification and regression tasks. Unlike many other algorithms, KNN does not make assumptions about the underlying data distribution and instead stores the entire training dataset in memory. When making predictions for a new data point, KNN identifies the k closest instances (neighbors) based on a chosen distance metric, such as Euclidean distance or cosine similarity. For classification tasks, it predicts the label of the new data point by majority voting among its nearest neighbors, while for regression tasks, it predicts the target value by averaging the target values of its nearest neighbors. However, the choice of the hyperparameter k is crucial, as smaller values increase model flexibility but may lead to overfitting, while larger values reduce the risk of overfitting but may increase bias.

6) CountVectorizer

CountVectorizer is a fundamental component in natural language processing (NLP), widely used for converting text data into numerical feature vectors. This technique plays a crucial role in various NLP tasks, such as document classification, sentiment analysis, and text clustering. CountVectorizer operates through several key steps. First, it tokenizes the input text, breaking it down into individual words or terms. Then, it constructs a vocabulary containing all unique terms present in the corpus of text documents. Each term in the vocabulary becomes a feature, and its position determines its index in the feature vector. Next, CountVectorizer counts the frequency of each term in each document, creating a document-term matrix. In this matrix, rows represent documents, columns represent terms, and each element denotes the count of occurrences of a term in a document. Finally, CountVectorizer transforms the text documents into numerical feature vectors, where each document is represented as a vector of term frequencies. This vectorization process facilitates the application of machine learning algorithms to text data, enabling tasks like classification and clustering. CountVectorizer offers flexibility through parameter tuning, allowing for adjustments such as n-gram range selection, stop word removal, and maximum document frequency, which can enhance feature extraction and model performance.

III. DIAGRAM OF MODEL

Database



Recommendations

IV. CONCLUSION

In conclusion, this research presents a comprehensive study on the development and implementation of a resume-based recommendation system utilizing natural language processing (NLP) techniques. The system aims to address the challenges faced by job seekers, particularly students, in navigating the competitive job market and effectively customizing their resumes to match job requirements. By leveraging advanced NLP models and deep learning algorithms, the system accurately extracts and analyzes information from unstructured resumes, facilitating semantic analysis and customization to specific industries and organizational standards. The integration of optical character recognition (OCR) and NLP enables seamless text parsing and analysis, while techniques such as stop word removal and lemmatization enhance data quality and relevance. Additionally, the TF-IDF method and cosine similarity calculation enable precise matching between job descriptions and candidate resumes, improving the efficiency and effectiveness of the resume-to-job matching process. Furthermore, the inclusion of K-Nearest Neighbors (KNN) classification offers a versatile approach to predicting candidate suitability based on similarity to job requirements. Overall, this research contributes to the advancement of resume parsing and recommendation systems, offering dynamic tools to enhance job seekers' visibility to recruiters and applicant tracking systems, ultimately improving the efficiency and success rate of the hiring process in today's digital era.

LITERATURE SURVEY

[1] S. P. Warusawithana, N. N. Perera, R. L. Weerasinghe, T. M. Hindakaraldeniya and G. U. Ganegoda, "Layout Aware Resume Parsing Using NLP and Rule-based Techniques," 2023 8th International Conference on Information Technology Research (ICITR), Colombo, Sri Lanka, 2023, pp. 1-5, doi: 10.1109/ICITR61062.2023.10382773. keywords: {Deep learning; Navigation; Resumes; Layout; Refining; Data mining; Surges; Layout aware; Resume Parser; Text extraction; NLP; Rule-based Techniques},

[2] R. Nimbekar, Y. Patil, R. Prabhu and S. Mulla, "Automated Resume Evaluation System using NLP," 2019 International Conference on Advances in Computing, Communication and Control (ICAC3), Mumbai, India, 2019, pp. 1-4, doi: 10.1109/ICAC347590.2019.9036842. keywords: {Resumes; Companies; Recruitment; Data mining; Databases; Task analysis},

[3] T. M. Harsha, G. S. Moukthika, D. S. Sai, M. N. R. Pravallika, S. Anamalamudi and M. Enduri, "Automated Resume Screener using Natural Language Processing(NLP)," 2022 6th International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2022, pp. 1772-1777, doi: 10.1109/ICOEI53556.2022.9777194. keywords: {Machine learning algorithms; Resumes; Education; Decision making; Companies; Market research; Natural language processing; NLP; Resume Screening; Hiring Process; Skill Set},

[4] N. Agrawal and A. Kaur, "An Algorithmic Approach for Text Recognition from Printed/Typed Text Images," 2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2018, pp. 876-879, doi: 10.1109/CONFLUENCE.2018.8442875. keywords: {Character recognition; Text recognition; Image recognition; Optical character recognition software; Image segmentation; Transforms; Image restoration; OCR; Otsu's algorithm; Hough transform; English alphabets; skew detection},

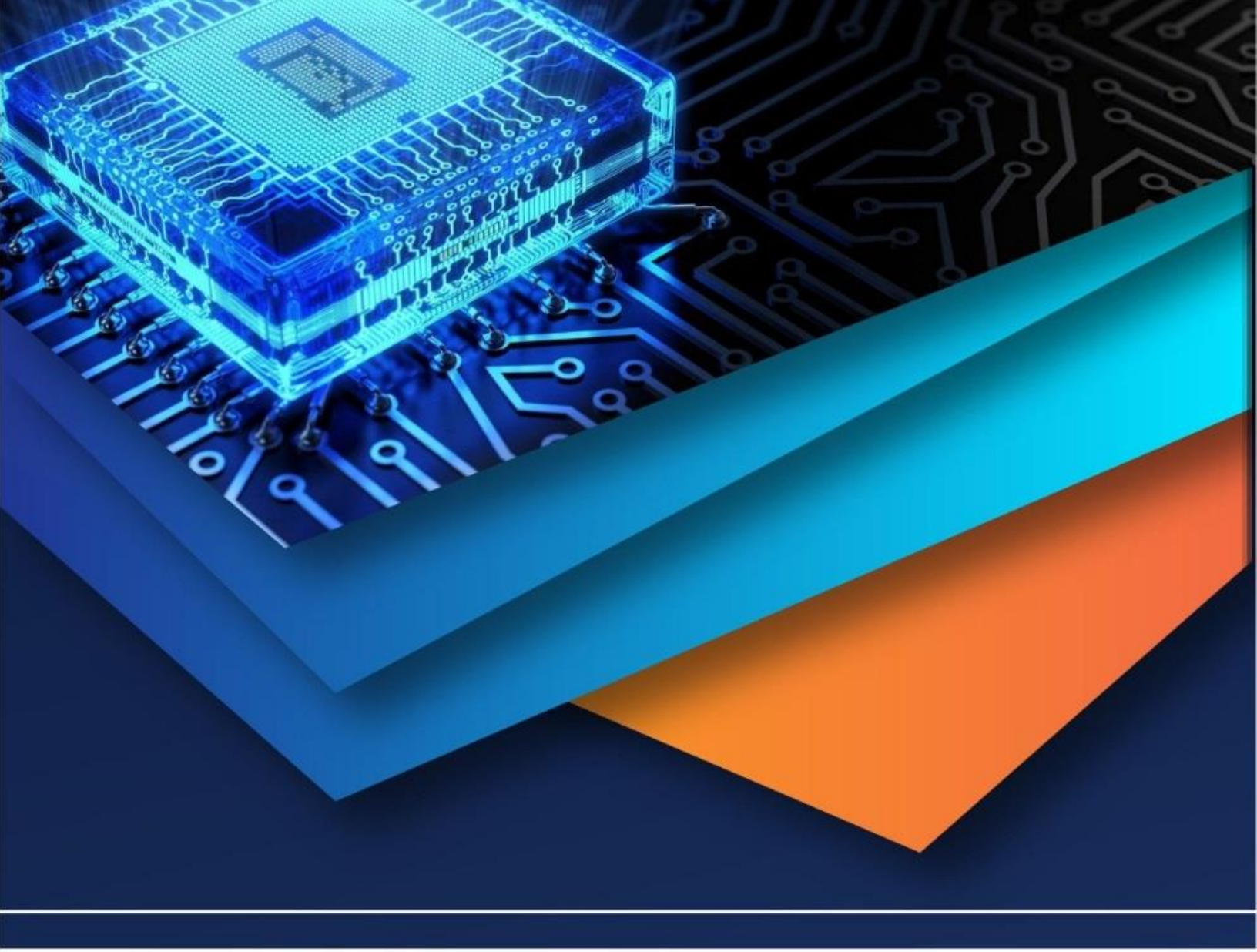
[5] S. Mhatre, B. Dakhare, V. Ankolekar, N. Chogale, R. Navghane and P. Gotarne, "Resume Screening and Ranking using Convolutional Neural Network," 2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS), Coimbatore, India, 2023, pp. 412-419, doi: 10.1109/ICSCSS57650.2023.10169716. keywords: {Support vector machines; Filtering; Computational modeling; Resumes; Manuals; Probability distribution; Convolutional neural networks; Convolutional Neural Network (CNN);Long Short-Term Memory (LSTM);Term Frequency - Inverse Document Frequency(TF-IDF);Cosine Similarity; Text Vectorization},

[6] A. Mankawade, V. Pungliya, R. Bhonsle, S. Pate, A. Purohit and A. Raut, "Resume Analysis and Job Recommendation," 2023 IEEE 8th International Conference for Convergence in Technology (I2CT), Lonavla, India, 2023, pp. 1-5, doi: 10.1109/I2CT57861.2023.10126171. keywords: {Machine learning algorithms; Resumes; Companies; Machine learning; Predictive models; Prediction algorithms; Natural language processing; Resume; Cosine similarity; Naïve Bayes; Web scraping; Job Recommendation},

[7] R. Bathija, V. Bajaj, C. Megnani, J. Sawara, and S. Mirchandani, "Revolutionizing Recruitment: A Comparative Study Of KNN, Weighted KNN, and SVM-KNN for Resume Screening," in 2023 8th International Conference on Communication and Electronics Systems (ICCES), 2023: IEEE, pp. 834- 840.

[8] B. Nisha, V. Manobharathi, B. Jeyarajanandhini and G. Sivakamasundari, "HR Tech Analyst: Automated Resume Parsing and Ranking System through Natural Language Processing," 2023 2nd International Conference on Automation, Computing and Renewable Systems (ICACRS), Pudukkottai, India, 2023, pp. 1681-1686, doi: 10.1109/ICACRS58579.2023.10404426. keywords: {Industries; Renewable energy sources; Resumes; Organizations; Real-time systems; Data mining; Recruitment; Natural Language processing; Resume Parsing and Ranking candidates},

[9] D. De, R. Dwivedi and N. Allwani, "Combined Application of Various Techniques for Personalized Job Recommendation," 2023 International Conference on Artificial Intelligence and Knowledge Discovery in Concurrent Engineering (ICECONF), Chennai, India, 2023, pp. 1-7, doi: 10.1109/ICECONF57129.2023.10083944. keywords: {Measurement; Urban areas; Employment; Organizations; Euclidean distance; Knowledge discovery; Concurrent engineering; Job Recommendation; Tanimoto; Jaccard; City Block; Manhattan; Cosine; Orchini; similarity measures; vectorization; preprocessing; accuracy},



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY