IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 1, August 2024

Job Analista : A Smart Resume Analyser and Recommendation System

Ronak Surve, Noel Monteiro, Sameer Shaikh, Shainila Shaikh
Don Bosco Institute of Technology, Mumbai, India
220ronak0028@dbit.in, 220noel0011@dbit.in, 220sameer0044@dbit.in, shainila@dbit.in

Abstract: *Machine learning, a subfield of data science, focuses on creating algorithms capable of learning from data and making predictions. Presently, recommendation frameworks are employed to address the issue of information overload across various domains, allowing users to focus on pertinent data. One area where such recommender systems can be invaluable is in aiding college graduates in finding employment opportunities aligned with their skill sets. Despite the abundance of websites offering extensive job information, this task remains daunting for students, who must sift through large volumes of data to identify suitable positions. Moreover, many students struggle to determine which jobs suit their abilities. In the thriving IT sector, engineering students are acquiring technical skills through courses, yet they lack guidance on which skills correspond to specific job roles. Existing job recommendation systems typically consider only the user's stated interests, overlooking their individual profile and skill set. This paper proposes a solution that examines users' resumes, evaluating their educational background, soft skills, hard skills, and project experience to provide tailored job recommendations. The system not only suggests suitable jobs but also assigns a score to the user's resume for each recommended position. Additionally, it offers recommendations for skill improvement to enhance the user's resume score.*

Keywords: Machine Learning, Recommendation Systems, Job Matching, Job Recommendations

I. INTRODUCTION

In the contemporary job market, the sheer volume and complexity of available data pose significant challenges for job seekers. Analista represents a pioneering solution, leveraging advanced methodologies to extract actionable insights from massive datasets. Rooted in data science and machine learning, our platform's primary objective is to offer precise and personalized job recommendations.

Every year millions of students are introduced to the actively competitive environment of job seekers. Finding a job that not only pays well but also meets your requirements is a tedious process, which requires countless hours of research from numerous resources across the Internet. There are many sources that are to be considered simultaneously, most of which are outdated or biased. In this situation knowing which sources to trust and getting an unbiased yet knowledgeable understanding of the Jobs is of utmost importance. With the growth in the Job market there's also been limitless expansion in the data which connects several technical and non-technical, personality based, experience based, skill set based, attitudebased parameters to each other to create thousands of large high complexity datasets which are impossible for the common person to analyze and benefit from.

This is where machine learning algorithms and data visualization techniques play a significant role in not only analyzing thousands of datasets but also filtering out the outliers and identifying a trend which it then depicts in simplified forms for the user to understand.

We aimed to minimize the time spent researching a job role by studying several resources, analyzing them and generating Machine learning models to identify patterns and provide an unbiased, relevant analysis for the same. By doing this we not only make it significantly easier for the user to find Jobs relevant to their interests, but only give them an overall understanding of how to increase their chances of acquiring that Job position.

DOI: 10.48175/568

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 1, August 2024

II. LITERATURE REVIEW

Machine learning is a field that has undergone tremendous research and various aspects of it have been studied in immense detail. In [1] a mobile application was developed for job applicants where we observe a machine learning model that uses content based filtering for Job recommendation, however, upon further study it was found that collaborative filtering could be a more effective future prospect for the same task as it compares similarities between users. This research paper[2] emphasizes about how Machine learning models are susceptible to bias because of prejudice in the training datasets which can cause unnecessary biasing in the end result. It also mentioned various methods that could be incorporated to minimize such biasing.

The paper [3] focuses on improving recommendation systems by highlighting out the limitations of traditional methods such as content filtering and collaboartive filtering. It proposes a deep learning approach to improve the quality of recommendation systems. This paper[4] also addresses the limitations of traditional job recommendation algorithms such as cold start and data scarcity for users with limited preferences, by proposing a hybrid appoach of combining deep learning techniques with collaborative filtering and content-based filtering. The paper [5] describes a proposed approach,DiffCF, addresses the challenge of building accurate recommender systems by utilizing differential machine learning. This method incorporates twin networks and differential labels, improving accuracy and training efficiency, even with small datasets. It presents empirical evidence demonstrating the superiority of DiffCF over standard collaborative filtering methods. The paper[6] proposes a job recommendation system by combining machine learning, NLP, and collaborative filtering. Addressing drawbacks in traditional systems, it incorporates skills matching, personality fit, job satisfaction, and retention probability. The approach enhances accuracy, but further improvements, like assessing cognitive abilities, are suggested for refining the system's understanding of applicants.

This paper[7] introduces a novel job recommendation approach that incorporates candidates' job preferences over time to handle the dynamic job market. The method combines machine learning (specifically Bidirectional Long Short Term Memory Networks with Attention) with non-machine learning strategies to generate diverse and exciting job recommendations. It addresses issues like generating serendipitous recommendations and solving the cold-start problem for new jobs and candidates. The approach utilizes latent competencies derived from skills, overcoming limitations of traditional skillbased methods. Empirical results show a significant increase in click-through rates (64.2%) on their job website, demonstrating the effectiveness of the proposed method.

In this paper[8], a job recommender system was developed to assist job seekers in finding suitable job opportunities based on their resumes. The system utilized Natural Language Processing and Machine Learning techniques, including Na¨ıve Bayes algorithm for resume classification and cosine similarity for skill matching. The system scraped job data from naukri.com and ranked the jobs based on their relevance to the user's skills.

III. SYSTEM DESIGN

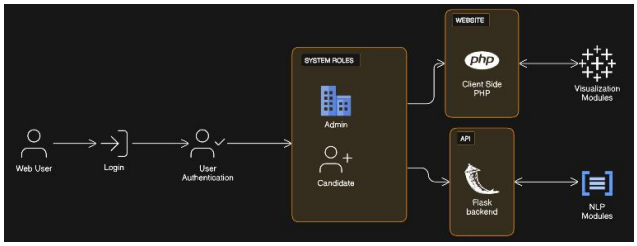


Fig. 1.System Architecture

The user diagram illustrates the seamless flow of interactions within the Analista job recommendation platform, highlighting the collaboration between three main components: PHP Frontend, PHP Backend, and Flask Application. The journey begins with the PHP Frontend, which hosts the user interface, providing an engaging and user-friendly experience. Within this interface, users encounter various links and buttons that trigger HTTP requests directed towards the Flask Application.

Upon activation, these HTTP requests are processed through the PHP Backend, where the Registration Form Processing Logic ensures the secure handling of user registration. The Registration module orchestrates the user authentication process, interfacing with the SQLite Database for data storage and retrieval. Following successful

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 1, August 2024

authentication, users are directed to the User Dashboard, where their information is displayed through the Display User Info module. This seamless transition between frontend and backend components ensures a fluid and secure user registration and authentication experience.

Additionally, the system incorporates a dedicated Resume Uploader module, which facilitates the upload of resumes by users. The Upload Processing Logic within this module is responsible for extracting relevant information from uploaded resumes. The extracted data, particularly skills, becomes a crucial input for the subsequent stages of the job recommendation process.

The Skills Extraction and Matching component refines the job recommendation process by extracting skills from resumes and employing them in generating personalized job recommendations. This step ensures that the recommendations align closely with the user's skill set and preferences.

The final stage takes place within the Job Recommendations Generation module, housed in the Flask Application. This module utilizes sophisticated algorithms and data processing techniques to generate personalized job recommendations based on user profiles and skills. The recommendations are then displayed to users through the Display Recommendations module on the PHP Frontend, completing the user journey with a comprehensive and tailored job recommendation experience. The synergy between frontend and backend components, facilitated by the Flask Application, ensures a dynamic, responsive, and user-centric platform for job seekers.

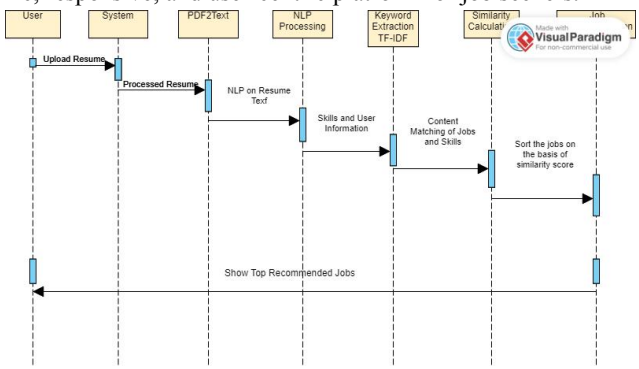


Fig. 2.Sequence Diagram

The system architecture is meticulously designed to provide an effective and seamless experience for users navigating the Analista job recommendation platform. At its core, the architecture comprises three primary components: PHP Frontend, PHP Backend, and Flask Application. The PHP Frontend encompasses the user interface, offering an intuitive and interactive design. It includes links and buttons that, when activated, initiate HTTP requests directed towards the Flask Application. On the PHP Backend side, the User Auth Module handles secure user authentication, ensuring a protected communication channel with the Flask Application. The User Dashboard module manages the display of user information and interacts with the Flask Application to fetch relevant data for a personalized user experience.

The Flask Application serves as the central hub of the system, hosting the Job Recommender functionality. It receives HTTP requests from the PHP Frontend, processes them, and interacts with the SQLite Database, which stores essential data related to user profiles, job recommendations, and more. Acting as both a web server and the communication interface, the Flask Application orchestrates the flow of information within the system. Simultaneously, the PHP Backend, which includes the Receive Recommendations and Display to User module, handles the retrieval of job recommendations from the Flask Application and presents them to users through the PHP Frontend.

In addition to these core components, the system features specialized modules like the Resume Uploader, which incorporates Upload Processing Logic for extracting pertinent information from uploaded resumes. The extracted skills, along with other relevant data, contribute to the subsequent stages of the job recommendation process. The Skills Extraction and Matching component further refines the job recommendation process by extracting skills from resumes and employing them in the generation of personalized job recommendations.

The overarching goal of the architecture is to streamline the user experience, from the initiation of actions on the PHP Frontend to the processing of requests on the Flask Application and the presentation of personalized job

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 1, August 2024

recommendations. By integrating PHP and Flask, the system achieves a harmonious synergy, leveraging the strengths of each component to provide users with a dynamic and responsive job recommendation platform.

The Job Analista project is built upon a Flask-based web application framework, employing Python for seamless HTTP request handling, HTML template rendering, and overall web application management.

Within the project, distinct components suggest diverse functionalities. A module responsible for algorithmic computations reflects the core of job recommendation strategies. Another component focuses on clustering techniques, likely used for grouping similar jobs or candidates.

One part of the system manages company-related information, while a web crawler is employed to retrieve data from online sources. There is a script dedicated to generating or preprocessing job-related data.

Other components indicate the generation of a conclusive web page and a corresponding CSV file, representing the output or presentation layer of the application. Integration with the Glassdoor API is suggested for retrieving additional job-related information. Modules for managing job data and reading PDF files, along with functionalities for parsing data in JSON format, contribute to the diverse set of features. Files related to reading PDF and CSV files imply data input capabilities, including resumes.

The system encompasses functionalities for student profiles, emphasizing a user-centric aspect. Additionally, data storage mechanisms, such as storing names, play a role in user profiles or data generation.

The project demonstrates a rich array of capabilities, including web scraping, data processing, algorithm implementation, and integration with external APIs. The components work collaboratively to achieve the overarching objective of the job recommendation system. A comprehensive diagram illustrating the interactions between these components would provide a visual representation of the data flow and processes within the system.

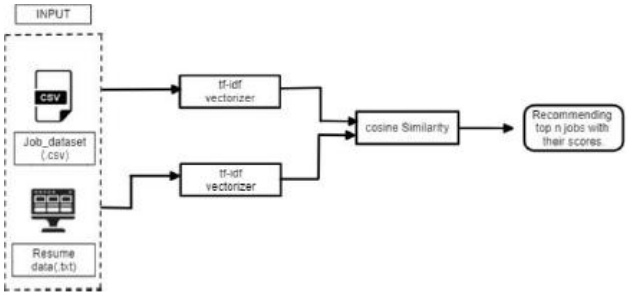
IV. METHODOLOGY

4.1 Data Preprocessing

Before training the machine learning models, a meticulous data preprocessing phase was conducted. This involved cleaning the collected data, handling missing values, and ensuring the data's uniform format. Additionally, measures were taken to address potential biases in the dataset, enhancing the fairness and reliability of the recommendation system.

4.2 Similarity Function Module

A user-provided resume in text format and a preprocessed employment dataset constitute the core of the Similarity Function Module depicted in Figure 3.4. The resume validation feature ensures that only (.txt) files are visible to the user when selecting a file. The algorithm selected for this process is cosine similarity with a TF-IDF vectorizer



Both the user's resume and the cleaned job dataset are input into the cosine similarity function, which undergoes preprocessing steps including stopwords removal and Porter stemming, consistent with the analysis of the job dataset's descriptions. The processed text files are treated as resumes, and TF-IDF is applied to both the resume and job description. This results in a matrix of values where terms appearing more frequently in a document are weighted higher by TF-IDF, balanced by their frequency across the entire corpus.

The cosine similarity formula, which computes the cosine of the angle between two vectors, is then employed using the matrix values. It involves calculating the dot product of the vectors and dividing it by the product of their lengths. The

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 1, August 2024

operational algorithm generates scores indicating the degree of similarity between the job description and the resume. The order of these scores is reversed to display the top-rated jobs first, allowing users to quickly focus on them.

4.3 Skills Recommending Module

The Skills Recommending Module, as illustrated in Figure 1, showcases the system's workflow, utilizing resumes and a skill dataset to determine the top five skills for each position. The subsequent paragraphs delve into the implementation specifics.

This module takes the skill dataset and resumes as inputs. The skill dataset contains skills required for various positions. The algorithm compares the word lists from both the skill dataset and the resume. Additionally, it prints the top five skills that do not match the resume. The system then presents the top 5 occupations and the top 5 skills that the resume holder needs to develop. Several cleaning procedures are applied to the skill dataset, such as removing prefix and suffix spaces to standardize the words.

V. PSEUDOCODE AND ALGORITHM

5.1 DATA PREPROCESSING

The preprocessing module cleans the job description data and the resume data, producing cleaned versions of both. The algorithm is outlined in Algorithm 4.1.

Input: Job dataset (.csv file)

Output: Cleaned Job dataset (.csv file)

Algorithm 5.1 Preprocessing

Read the job dataset file.
Use pandas to read the file and store it in a variable, say 'k'.
Separate the job and description attributes from 'k' using Python's split operation.
Convert the description into a list of words.
For each word in the description:
If the word is not a stopword:
Apply Porter stemming to the word.
Convert the list of stemmed words into a text file.
Otherwise, discard it.

5.2 SIMILARITY FUNCTION MODULE

The similarity function module compares the user's resume with the cleaned job descriptions to recommend top jobs along with their corresponding scores. The algorithm is described in Algorithm 5.2.

Input: Cleaned job dataset and resume

Output: N recommended jobs in descending order of scores

Algorithm 5.2 Similarity Function Module

Upload the resume.
If the resume is uploaded:
Read and store the resume in a folder.
Convert the resume into a list, removing spaces, tabs, and empty elements.
Apply preprocessing to the resume using the preprocessing module.
Apply the cosine similarity formula to both the job descriptions and the resume.
Convert the matrix into an array, with the job description array as 'arr1' and the resume array as 'arr2'.
For each element 'l1' in 'arr1':
For each element 'l2' in 'arr2':
Calculate the cosine similarity score 's1' and 's2' using the formula.
Append the score to a list.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 1, August 2024

Sum the duplicate job scores, store them in a dictionary, and calculate their average.
Sort the scores in descending order and display the results.
The main functionality of the project is implemented in Algorithm 5.2, with only the skill recommendation module remaining.

5.3 SKILLS RECOMMENDING MODULE

This module recommends the top 5 skills for the top 5 recommended jobs to the user for self-improvement. The algorithm is outlined in Algorithm 5.3.

Input: Processed resume and skill dataset (.csv file)
Output: Top 5 skills for each of the top 5 jobs

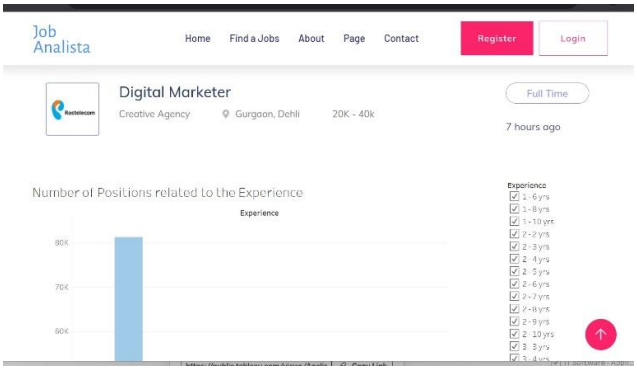
Algorithm 4.3 Skills Recommending Module

Read the skill dataset file.
Store the file using pandas in a variable, 'k'.
Separate the job titles and corresponding skills, and zip them together.
Select the top 5 jobs from the output of the similarity function module and save them in a list.
For each job in the top 5 list:
Retrieve the corresponding skills from the skill dataset.
For each skill:
If the skill is not already present in the user's resume:
Store it in a recommended skills list.
Otherwise, ignore it.
Slice the recommended skills list to include only the top 5 skills.
Display the job titles along with the recommended skills for improvement.

VI. IMPLEMENTATION AND RESULTS

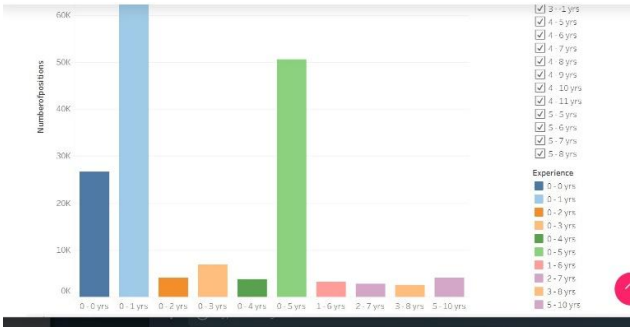
The Job Analista project leverages a combination of cutting-edge technologies and programming languages to build a robust and dynamic job recommendation system. The frontend of the system is developed using PHP, a versatile and widely-used scripting language known for its simplicity and efficiency in web development. PHP provides a seamless user interface, incorporating interactive elements and ensuring a smooth experience for users interacting with the platform. The frontend serves as the gateway for users to access various functionalities offered by Analista.
On the backend, PHP is employed for handling the registration form and its processing logic. The choice of PHP for these components is driven by its server-side scripting capabilities, making it well-suited for processing user inputs, performing necessary computations, and communicating with the Flask application. The PHP backend also manages user authentication, ensuring secure and user-friendly access to Analista's features.
The Flask application serves as the core of the system, facilitating the job recommendation process. Flask, a lightweight and versatile web application framework for Python, is selected for its simplicity, scalability, and ease of integration with machine learning models. The Flask framework enables seamless HTTP request handling, HTML template rendering, and efficient overall management of the web application. Furthermore, Flask acts as the communication bridge between the PHP backend and the SQLite database, ensuring a cohesive and well-integrated system.
The choice of SQLite as the database for Analista is deliberate and aligns with the project's requirements. SQLite is a lightweight, file-based relational database management system that is suitable for smaller-scale applications. In the context of Analista, SQLite efficiently stores and retrieves data, contributing to the responsiveness and agility of the recommendation system. Its ease of integration with Flask simplifies data management tasks, and its performance characteristics align with the anticipated scale of the project.
Data Visualization:

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 1, August 2024



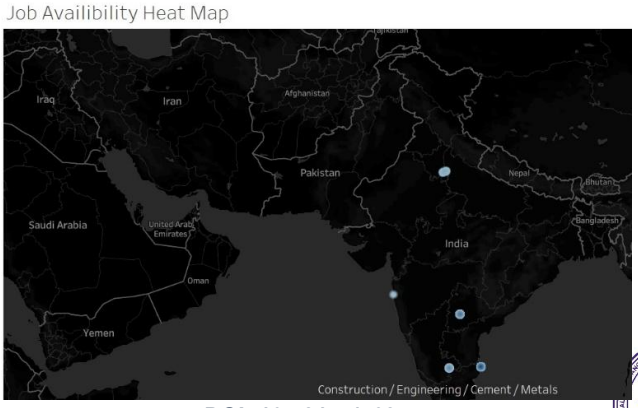
In the implementation, data visualization tools such as Tableau were strategically employed to augment user comprehension of employment trends. These visualizations provide a dynamic and insightful representation of complex datasets, offering users a clearer understanding of the job market landscape. The integration of Tableau facilitates the presentation of information in a visually engaging manner, contributing to a more user-friendly experience.

A variety of visualizations were created to showcase different aspects of employment trends. These may include charts, graphs, and interactive dashboards that highlight key metrics, patterns, and correlations within the dataset. For instance, a graph could depict the distribution of job opportunities across various industries, while a heatmap might illustrate the geographical concentration of job openings.

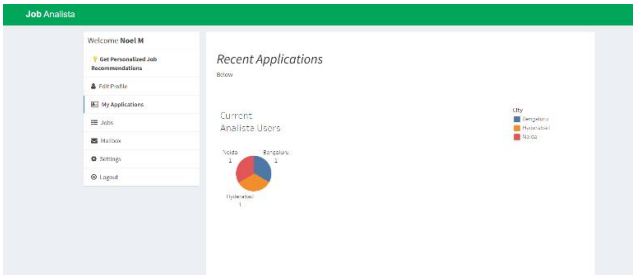


To provide a tangible example, consider a Tableau dashboard displaying the demand for specific skills in the job market. The dashboard could feature interactive elements allowing users to explore skill trends over time, identify emerging skills, and compare the demand for different skill sets. This dynamic visualization not only enhances user engagement but also empowers job seekers and employers with actionable insights.

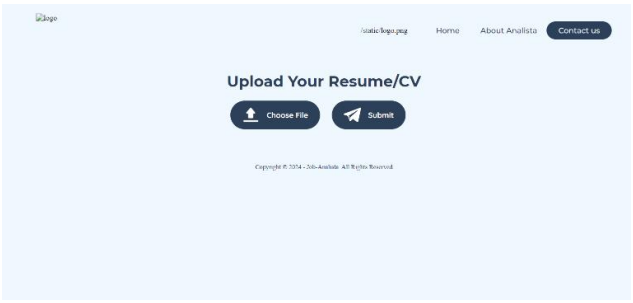
For a comprehensive understanding, screenshots of these Tableau visualizations can be included in the documentation. These screenshots should be strategically selected to represent the diversity of visualizations created and their relevance to the overall analysis. Including captions or annotations with each screenshot will further elucidate the significance of the visual representation, aiding users in interpreting the employment trends effectively.



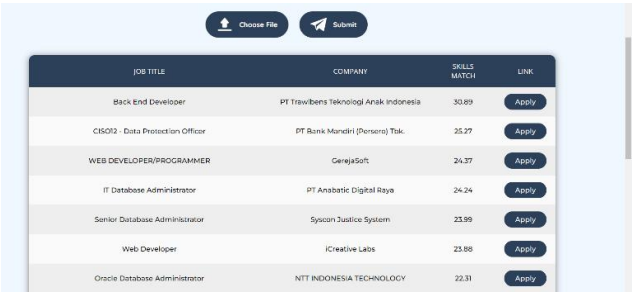
IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 1, August 2024



The user dashboard in the provided screenshot showcases a well-designed interface within the Job Analista platform. It includes essential features such as personalized job recommendations, an option to edit user profiles, a section for managing job applications, a mailbox for communication, and standard account settings. The "Recent Applications" area enables users to conveniently track their recent job submissions, facilitating efficient application management. The platform emphasizes user experience by tailoring job suggestions to individual profiles and providing a centralized hub for users like Noel M to navigate and optimize their job search journey.



The "Job Recommendation System" page simplifies the user experience with a prominent "Choose File" button for resume uploads. Users like Noel M can easily obtain personalized job suggestions by clicking "Get Recommendations," highlighting the platform's user-friendly approach to job matching.



The "Job Recommendations" page offers a clear breakdown of recommended roles and locations based on user skills. For Noel M, whose skills include Python, Java, JavaScript, HTML, CSS, Problem Solving, SQL, Machine Learning, Data Visualization, React, PHP, and Bootstrap, the system suggests various roles in different locations. Some highlighted recommendations include positions like "Python Developer" in Chennai, "Java Full Stack Developer" in Pune, "Senior DevOps Engineer" in Chennai, and others. Each role is accompanied by details such as job salary, required experience, and location, enhancing transparency for users like Noel to make informed decisions about potential job opportunities.

VII. CONCLUSION

Job Analista emerges as a cutting-edge solution in the complex realm of the modern job market. Leveraging advanced data science and machine learning methodologies, it redefines the job-seeking experience by providing users with highly personalized and relevant job recommendations. The integration of collaborative filtering, deep learning, and adaptive learning techniques ensures precision in matching users with suitable opportunities. The multidimensional

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 1, August 2024

approach, including skills extraction, real-time adaptability, and responsive studies, establishes Job Analista as a dynamic and reliable companion in the ever-evolving job market landscape. The user-friendly interface and incorporation of data visualization tools contribute to an enhanced understanding of employment trends. As showcased through the user dashboard and job recommendation system, Job Analista not only simplifies the job search process but also adapts and grows with user interactions. Ultimately, Job Analista stands at the forefront of innovation, providing a comprehensive and effective solution to navigate the complexities of the modern job market.

VIII. FUTURE SCOPE

Job Analista holds great promise as it focuses on continuous improvement and innovation. To enhance the precision of job recommendations, the platform will invest in advanced machine learning models, incorporating cutting-edge techniques in natural language processing and reinforcement learning. Additionally, exploring the integration of emerging technologies like blockchain for credential verification and augmented reality for an enriched user experience is on the agenda. The platform aims for global expansion by collaborating with international job boards, fostering a more diverse and comprehensive job database. To further engage users, Job Analista plans to introduce interactive features such as real-time chat support and virtual career fairs. Moreover, it envisions extending its role beyond job recommendations by incorporating skill development and training modules, potentially partnering with online learning platforms to offer tailored courses and certifications. These strategic initiatives position Job Analista at the forefront of providing a dynamic and holistic approach to job seeking and career development.

ACKNOWLEDGMENT

We thank our colleagues and the institute for their assistance with this project. This research would not have been possible without the help of peers and experts who took the time to share their thoughts on previous research editions of this paper. We also thank our family and friends who have provided support and inspiration throughout our academic careers.

REFERENCES

[1]. S. Gadegaonkar, D. Lakhwani, S. Marwaha and P. A. Salunke, ”Job Recommendation System using Machine Learning,” 2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS), Coimbatore, India, 2023, pp. 596-603, doi: 10.1109/ICAIS56108.2023.10073757.

[2]. J. Kavitha, J. S. Kiran, S. D. V. Prasad, K. Soma, G. C. Babu and S. Sivakumar, ”Prediction and Its Impact on Its Attributes While Biasing MachineLearning Training Data,” 2022 Third International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE), Bengaluru, India, 2022, pp. 1-7, doi: 10.1109/ICSTCEE56972.2022.10100010.

[3]. B. Dellal-Hedjazi and Z. Alimazighi, ”Deep learning for recommendation systems,” 2020 6th IEEE Congress on Information Science and Technology (CiSt), Agadir - Essaouira, Morocco, 2020, pp. 90-97, doi: 10.1109/CiSt49399.2021.9357241.

[4]. W. Chen, X. Zhang, H. Wang and H. Xu, ”Hybrid deep collaborative filtering for job recommendation,” 2017 2nd IEEE International Conference on Computational Intelligence and Applications (ICCIA), Beijing, China, 2017, pp. 275-280, doi: 10.1109/CIAPP.2017.8167222.

[5]. M. N. Abadeh, Z. Derakhshandeh and M. Mirzaie, ”An Efficient Collaborative Filtering for Recommendation Systems Using Differential Machine Learning,” 2022 9th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS), Bam, Iran, Islamic Republic of, 2022, pp. 1-7, doi: 10.1109/CFIS54774.2022.9756424.

[6]. K. Appadoo, M. B. Soonnoo and Z. Mungloo-Dilmohamud, ”Job Recommendation System, Machine Learning, Regression, Classification, Natural Language Processing,” 2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), Gold Coast, Australia, 2020, pp. 1-6, doi: 10.1109/CSDE50874.2020.9411584.

[7]. Nigam, A. Roy, H. Singh and H. Waila, ”Job Recommendation through Progression of Job Selection,” 2019 IEEE 6th International Conference on Cloud Computing and Intelligence Systems (CCIS), Singapore, 2019, pp. 212-216, doi: 10.1109/CCIS48116.2019.9073723.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 1, August 2024

[8]. Mankawade, V. Pungliya, R. Bhonsle, S. Pate, A. Purohit and 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.

[9]. G. K. Gupta, R. S. Rathore, P. S. Rajput, and R. Jain, "Enhancing Job Recommendations using Ensemble Learning," 2021 International Conference on Information Technology (ICIT), Jaipur, India, 2021, pp. 1-5, doi: 10.1109/ICIT51192.2021.00088.

[10]. S. Malhotra, S. Choudhury, R. Verma, and P. Arora, "A Comparative Analysis of Job Recommendation Algorithms in E-Recruitment Systems," 2018 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE), Phagwara, India, 2018, pp. 1-5, doi: 10.1109/ICCIKE.2018.8692977.

[11]. R. S. Choudhury, A. K. Sharma, S. K. Aggarwal, and S. A. Khan, "Personalized Job Recommendations Using Hybrid Filtering Techniques," 2019 4th International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS), Gwalior, India, 2019, pp. 1-6, doi: 10.1109/CSITSS.2019.8939856.

[12]. V. Mathur, S. S. Parihar, M. Bhatia, and V. Tiwari, "Job Recommendation in E-Recruitment using Deep Learning," 2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 2022, pp. 1-5, doi: 10.1109/ICSSIT53279.2022.9697484.

[13]. A. S. Kumar, M. K. Gupta, A. Sharma, and P. K. Sharma, "Context-Aware Job Recommendation System using Tensor Decomposition," 2020 4th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2020, pp. 837-842, doi: 10.1109/ICCMC51211.2020.9148056.

[14]. S. Chaturvedi, A. Jain, R. Thapar, and P. K. Sharma, "A Comprehensive Review on Job Recommendation Systems," 2021 8th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2021, pp. 1914-1918, doi: 10.1109/INDIACOM52020.2021.9370113.

[15]. K. K. Sharma, S. Saini, R. Garg, and S. K. Sharma, "Job Recommender System using Deep Learning and Sentiment Analysis," 2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE), Phagwara, India, 2019, pp. 1-5, doi: 10.1109/ICCIKE.2019.8899602.

[16]. V. Sharma, M. Saini, A. Agarwal, and P. K. Gupta, "Effective Job Recommendation System using Hybrid Filtering Techniques," 2022 IEEE 1st International Conference on Advances in Information Technology (ICAIT), Nagapattinam, India, 2022, pp. 148-153, doi: 10.1109/ICAIT.2022.9659396.

[17]. A. G. Patil, K. J. Pawar, R. G. Gawade, and S. A. Deshmukh, "Machine Learning-based Job Recommendation System," 2018 International Conference on Computer, Electrical and Communication Engineering (ICCECE), Pune, India, 2018, pp. 1-5, doi: 10.1109/ICCECE.2018.8787497.

[18]. N. Sharma, S. V. Shukla, P. Singh, and V. K. Singh, "A Comparative Analysis of Job Recommendation Algorithms," 2020 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS), Bhopal, India, 2020, pp. 1-5, doi: 10.1109/SCEECS49216.2020.9206415.

[19]. R. Kumar, S. A. Gadekar, and N. A. Palkar, "Job Recommendation System using Machine Learning Algorithms," 2019 IEEE 3rd International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2019, pp. 673-678, doi: 10.1109/ICECA.2019.8821923.

[20]. M. Singh, P. Sharma, N. Goyal, and R. Singh, "Job Recommendation System Using Hybrid Algorithm," 2018 3rd International Conference for Convergence in Technology (I2CT), Pune, India, 2018, pp. 1-5, doi: 10.1109/I2CT.2018.8529748