International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 02 | Feb 2026 www.irjet.net p-ISSN: 2395-0072

Career Guidance & Recommendation on Placement using Machine Learning and DSA Visualizer

Jaldu Bhupesh Sai Sri[1], Malees Greeshma[2], Maddi Sateesh Reddy[3]

[1]*Final Year Undergraduate Student, Dept. of Computer Science and Engineering, Joginpally B.R. Engineering College (JBREC), Hyderabad, Telangana, India*
[2]*Final Year Undergraduate Student, Dept. of Computer Science and Engineering, Joginpally B.R. Engineering College (JBREC), Hyderabad, Telangana, India*
[3]*Final Year Undergraduate Student, Dept. of Computer Science and Engineering, Joginpally B.R. Engineering College (JBREC), Hyderabad, Telangana, India*

Abstract - In the rapidly evolving landscape of software engineering, students face a dual challenge: mastering complex Data Structures and Algorithms(DSA) required for technical interviews and navigating an overwhelming array of career specializations. While existing solutions address these issues in isolation - either through standalone algorithm visualizers or static career counselling tools- there is a lack of integrated platforms that bridge the gap between skill acquisition and career discovery. This paper proposes a novel, dual module system. The first module is an interactive ***Algorithm Visualizer*** developed using React, capable of animating ***sorting, searching,*** *and* ***graph algorithms*** to enhance conceptual retention. The second module is an ***Ai-Job Recommendation Engine*** built on a Flask backend. It utilizes ***TF-IDF vectorization*** and ***Cosine Similarity*** on a dataset of over 1 million job records to match user skills with optimal job roles. Furthermore, the system integrates Generative Ai (via Groq/Llama-3) to create personalized learning roadmaps and aggregates course resources. The results demonstrate a high-precision matching capability and an improved learning experience for students preparing for technical placements.

Key Words: **Algorithm visualizer, sorting, searching, graph algorithms, Ai-Job Recommendation Engine, TF-IDF vectorization, Cosine Similarity**

1. INTRODUCTION

The demand for skilled software engineers has led to a proliferation of specialized roles, ranging from Data Science to Full Stack Development. However, a significant disconnects remains between the academic curriculum and industry requirements. Two primary hurdles exit for engineering students: the difficulty in visualizing abstract algorithmic concepts and the uncertainty in selecting a career path that aligns with their acquired skills.

Traditional teaching methods for algorithms often rely on static diagrams, which fail to convey the dynamic nature of operations like recursive tree traversals or graph pathfinding. Simultaneously, career guidance is often subjective, relying on human advisors who may lack real-time data on industry trends. This research paper presents a unified web-based platform that addresses these challenges holistically. The proposed system combines:

1.1 Visual Learning

A dynamic visualizer for arrays, linked lists, stack, queue, trees, graphs.

1.2 Intelligent Guidance

A content-based recommendation system that analyses user skills against a massive dataset of job descriptions to predict suitable roles.

1.3 Actionable Roadmaps

Integration of LLMs to generate step-by-step preparation guides for the recommended roles.

2. LITERATURE SURVEY

The development of this system draws upon various domains of educational technology and machine learning.

2.1 Placement Prediction Systems

Kulkarni et al. developed an algorithm visualizer using React.js, focusing on sorting and pathfinding algorithms like Merge Sort, Dijkstra's algorithm. Their research highlighted that visual information is processed faster than abstract text, significantly aiding student retention. However, their scope was limited to visualization without linking these skills to specific career outcomes.

2.2 Placement Prediction Systems

Divya et al. proposed a placement analysis system using supervised machine learning algorithms such as Support Vector Machines(SVM) and Random Forest. Their work focused on classifying students based on academic history(grades) to predict the probability of placement. While effective for administrative forecasting, it lacks a

mechanism to recommend specific job roles based on technical skills rather than just grades.

2.3 Personalized Career Recommendation

Qamhieh et al .introduced a fuzzy logic-based recommender system (PCRS) for engineering students. Their approach utilized personality tests and academic performance to suggest engineering disciplines. While novel, this approach relies heavily on psychometric data rather than the explicit technical skills and keywords that modern applicant Tracking Systems(ATS) and recruiters prioritize.

3. METHODOLOGY

The proposed system architecture is divided into two distinct but complementary modules: the Learning Module (Visualizer) and Recommendation Module (Career AI).

3.1 Module 1: DSA Visualizer

The visualizer is built using a modern JavaScript framework to render animations of data structures. It supports the following operations:

- **Searching:** Linear, Binary, Jump, Interpolation, Exponential, and Fibonacci search.
- **Sorting:** Visualization of bubble, insertion, quick, and merge sort allowing users to see element swapping and partitioning in real-time.
- **Linear Data Structures:** Linked Lists(Traversal, Insertion, Deletion), Stacks(push/pop), and Queues(Enqueue/Dequeue).
- **Non-Linear Data Structures:** Trees (BFS,DFS) and Graph(BFS/DFS).

This module prepares the user for the technical interviews associated with the jobs recommended by Module 2.

3.2 Module 2 : AI Job Recommendation Engine

This module utilizes a Content-Based Filtering approach. Unlike collaborative filtering, which requires user history, this system analyses the semantic relationship between a user's skill set and job descriptions.

1. **Data Collection:** A dataset containing approximately 1 million job records, including titles and skill requirements, is utilized.
2. **Preprocessing:** Job description and user inputs are cleaned (removing stop words, and special characters).
3. **Vectorization(TF-IDF):** We employ the Term Frequency-inverse Document Frequency (TF-IDF) vectorizer. The static is intended to reflect how important a word (skill) is to document (job description) in a collection or corpus.
4. **Similarity Calculation:** We calculate the Cosine similarity between the user's between skill vector (U) and the job vectors (J) to find the closest matches.

3.3 Generative Roadmap & Course Aggregation

Once a role is predicted, the system calls the Groq API(utilizing Llama-3 models) to generate a custom markdown roadmap. Simultaneously, a web scraper fetches relevant courses from educational platforms to provide immediate learning resources.

4.SYSTEM IMPLEMENTATION

The system implemented using a Python-Flask backend and React.js frontend.

4.1 Backend Logic (Python)

The core recommendation logic is encapsulated in train.py and app.py.

- **Training:** The train.py script reads the dataset in chunks to manage memory efficiency. It initializes a Tf-IDF Vectorizer with a vocabulary limit of 12,000 features to reduce noise. The resulting sparse matrix is cached using pickle for fast inference.
- **Inference:** In app.py, the get_recommendations function transforms user input into a vector and computes cosine similarity against the cached matrix.
- **Course Scrapping:** courses.py uses BeautifulSoup to scrape course data. It includes a fallback. Mechanism to serve curated course lists(e.g., for "Data Scientist" or "Web Developer") if live scraping fails or is blocked.

4.2 Generative AI Integration

The roadmap.py module integrates the Groq client . A prompt engineering approach is used to instruct the LLM to act as a "Career Coach," generating structured advice covering Foundation, Core Competencies, and Advanced Specialization.

4.3 Visualizer Implementation

The visualizer maintains the state of data arrays. As algorithms execute, state changes (swap, comparisons) are pushed to an animation queue, which is rendered sequentially to the user, providing a step-by-step walkthrough of the algorithm.

5. RESULTS AND DISCUSSION

5.1Recommendation Accuracy

The model was evaluated based on the sparsity of the TF-IDF matrix. With a sparsity greater than 96% , the model demonstrates a high ability to distinguish between

unique skill sets (e.g., distinguishing “React/Redux” for Frontend roles vs. “Pandas/Scikit-learn” for Data Science).

- **Input:** “python,SQL,React”
- **Output:** Full Stack Developer, Python Developer, Data Engineer.

The system successfully identifies roles that utilize the intersection of provided skills.

5.2 User Interface

The interface features a “premium dark UI” designed for visual comfort. The results section dynamically renders the “Match Strength” (High/Medium) based on the cosine similarity score, capped at 96% to avoid overfitting illusions.

5.3 Performance

- **Latency:** The use of cached Pickle model allows for sub-second recommendations(inference time < 200ms).
- **Scalability:** The chunking method used in training allows the system to handle datasets exceeding 10 Lakh rows without memory overflow errors.

6. CONCLUSION

This paper presented an integrated platform that merges algorithmic education with AI-driven career planning. By utilizing TF-IDF and Cosine Similarity, the system provides objective, skill-based job recommendations, overcoming the subjectivity of traditional counselling. The inclusion of the DSA visualizer ensures that students have the immediate resources to prepare for the technical demands of their recommended careers. Future work will involve integrating a collaborative filtering mechanism to recommend jobs based on successful alumni profiles and expanding the visualizer to include complex dynamic programming problems.

REFERENCES

[1] A. Kulkarni, S. Padave, S. Shrivastava, and V. Kawtikwar, "Algorithm Visualizer," International Journal for Research in Applied Science and Engineering Technology (IJRASET), vol. 11, no. VII, pp. 1818-1823, July 2023.

[2] N. Divya, S. Namburu, and R. Raja, "Student Placement Analysis using Machine Learning," in Proceedings of the 8th International Conference on Communication and Electronics Systems (ICCES 2023), IEEE, 2023, pp. 1027-1031.

[3] M. Qamhieh, H. Sammaneh, and M. N. Demaidi, "PCRS: Personalized Career-Path Recommender System for Engineering Students," IEEE Access, vol. 8, pp. 214039-214049, 2020.

[4] F. Provost and T. Fawcett, Data Science for Business, O'Reilly Media, 2013. (General ML Reference)

[5] J. Ramos, "Using TF-IDF to Determine Word Relevance in Document Queries," Proceedings of the First Instructional Conference on Machine Learning, vol. 242, pp. 29-48, 2003. (Methodology Reference)