ADROID Conference Series: Engineering and Technology
Volume 2, Issue 3, pp.19-29, 2026

AI-Driven Personalized Learning and Career Recommendation System

Riya Ranjan Jha, Sanjana Pattanayak, Tanisha
Department of Computer Science and Engineering (AI & ML), ABES, India
riya.22b1531137@abes.ac.in, sanjana.22b1531040@abes.ac.in, tanisha.22b15311042abes.ac.in

Abstract. Data Structures and Algorithms (DSA) is a crucial requirement in the competitive technology market, but existing material, such as Data Structures sheets, provides a one-size-fits-all solution. This deprivation of individuality will result in poor learning processes because students are unable to concentrate on their own areas of weaknesses. This is also evident in career navigation, as students struggle to align their qualifications with relevant job vacancies, resulting in a poor and unfocused application process. The proposed research paper provides an intelligent, automated platform that bridges this divide by providing tailored career suggestions and learning plans. The technology assesses the user's coding skills and combines content-based filtering with language model refinement to produce a unique study plan that targets the user's areas of weakness [5]. The technology examines such variables as time complexity and approach to provide automated analysis of code submissions. It also compares job vacancies relevant to the skills displayed by the user. The primary contribution of this research is the creation of a one-stop shop that will automatize personalized mentoring. We demonstrate how AI can be applied in a practical way to close the feedback loop in technical education, providing students with a more direct and efficient path to success in the software field.

Keywords: Artificial Intelligence, Large Language Models, Fine-tuning, Skill Assessment, Closed feedback loop, Skill-job matching, Personalized Job Recommender Systems, Personalized Roadmap Generation, AI-driven Code Analysis, Intelligent Tutoring System, Content-based Filtering

1. INTRODUCTION

Knowledge of Data Structures and Algorithms (DSA) is a vital factor in defining the ability to solve problems and is also one of the requirements of students who wish to work in the software industry [1]. Whilst the existing 'DSA sheets' are often utilized by the students to practice, such materials normally offer a generalized strategy that does not take into account the special conceptual shortages and learning needs of each learner and do not perform in line with the individual learner pace of learning. To overcome these limitations, the intelligent tutoring system proposed in this research article will provide career guidance and personalized learning plan. The program identifies the strengths and the weaknesses in a student by examining the user provided code with sophisticated language models to detect areas of weaknesses. The following is the creation of a dynamic DSA practice sheet: this AI-based, incremental engine and updated at regular intervals every 14 days according to the performance of the student in the regular tests. The system extends the practice of academic practice in that it matches skills exhibited by the user and suitable employment opportunities, and generates a industry standards which offer career preparation.

The key contribution of this work is that it provides one platform which gives students the opportunity to study at their pace and focus on a well thought out plan of study, which would be changed as they proceed. This paper reveals how AI and ML models can be applied in practice to provide the software engineering learners with a more effective and personalized learning path.

A. Problem Statement

The current materials available in learning Data Structures and Algorithms (DSA) are not modified to track the development of the individual learners. In the absence of clear instructions as to where to begin or even how to go about their studies at a slow pace, learners end up with a useless and disheartening learning process. Generalist AI tools such as ChatGPT, provide basic, but they tend to provide only a superficial correction to codes and cannot be used to construct an organized learning path dependent on the unique conceptual strengths and weaknesses of each user without a lot of human prodding [4]. The user must give a considerable number of follow-ups even when they attempt to do so. The absence of a dedicated AI mentor creates a large vacuum in the learning process. The problem goes further to career preparation, whereby the students are unable to develop career wise as the job hunt to gain employment which they are not suited in terms of skills is poorly attempted because of the inaccurate self-assessment of their talents.

B. Objectives

The following are the priorities of the objectives of this research:

- to build an intelligent, automated AI teacher that provides Data Structures and Algorithms (DSA) teaching to each student individually.
- to fine-tune a Large Language Model (LLM) that can generate a dynamic and personalized learning path based on the performance of a student.
- to implement a machine learning model which
- offers specific job opportunities by aligning the demonstrated skills of a student with relevant job descriptions.
- to incorporate a system of routine evaluations that keep track of the development of skills by a student that would give a history of his growth.
- to augment to give a service that takes into account the input and talents of the user to develop a resume that will be optimized to the Applicant Tracking System (ATS).
- to measure the performance of the recommendation and the corresponding models with the help of the widely used metrics such as F1-score, recall, accuracy, and precision.
- to open up the whole system to be usable as a web application so that users can gain access to a personal dashboard with which to see their progress.

Our approach would aim at using the huge volumes of historical information that is available and transform it into something useful to the society. The learning part will include a large set of DSA questions based on publicly available, industry-accepted practice sheets and coding sites. This data is used as question bank to personalized road maps and student tests. The career component will be sourced via numerous online listing platforms where the job descriptions will be collected. This data will be used to train the machine learning models that will be responsible of correctly and relevantly matching the student talents to the employment opportunities.

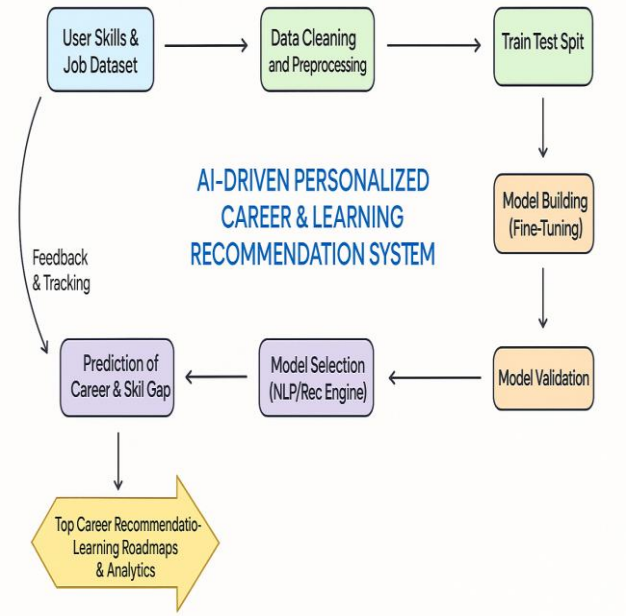

Figure 1: AI-Driven Personalized Carrer & Learning Recommendation System

2. RELATED WORK

Artificial Intelligence of personalized education and career guidance has increased significantly, and several platforms with data use are provided to enhance the results of users. Nowadays, these systems can be categorized into 3 main variables, namely, Massive Open Online Courses (MOOCs), interactive coding platforms, and professional networking sites. MOOCs such as Coursera and edX rely on recommendation engines that recommend courses to them depending on what they are viewing and what they indicate they are interested in [1]. These suggestions assist individuals in discovering something, and they do not necessarily highlight certain flaws in abilities of somebody. There are a lot of coding problems to practice on interactive coding platforms, such as LeetCode and HackerRank[1]. They however, have rather straightforward evaluation systems, in which they simply find out whether the answer is correct or incorrect, not assessing the quality, efficiency or writing ability of the code.

But their assessment processes are usually not complex, simply whether the answer is correct or not but not whether the code is good, efficient or well written. Instead, they are able to indicate whether a user has solved a problem, but not the quality of how they did it.Professional online networking systems such as LinkedIn have introduced functions such as skill endorsements recommendations [6]. The systems typically match keywords based on the profile of a user to job descriptions and endorsements with social validation

3. THEORY/CALCULATION

A. Vector Embeddings

To mathematically compare unstructured text such as user skills and job descriptions we first convert them to dense numerical representations also known as vectors or embeddings. This is done on the Vertex AI platform with the text-embedding-gecko model of Google. All of the user profiles and job listings are turned into a high-dimensional vector, and analogous meanings are put nearer to one another in the space of vectors.

B. K-Nearest Neighbors for Recommendation

In order to identify the most relevant job opportunities to a user in a large database, we employ the K-Nearest neighbors (KNN) algorithm. On receiving a user vector KNN algorithm identifies the nearest k closest job vectors within the job vector space [5]. The Google Cloud Vertex AI Matching Engine will handle this quick similarity search, and is configured to make fast and scalable Approximate Nearest Neighbor (ANN) searches on large datasets [3].

C. Large Language Model for Code Analysis

The Gemini Pro large language model on tasks that require a profound comprehension and capability of creating text that is natural and human-like[2]. On entering code by a user, a lengthy prompt is generated which contains the problem being solved and a solution provided by the user. This is then forwarded to the Gemini API[4]. The model is requested to critically review the code, provide feedback on its efficiency (time and space complexity, such as O(n)) and the code practices good coding practices, does it contain logical errors, and how the problem can be solved differently. This provides the users with useful and informative insights that are far more helpful than simple right or wrong answer.

D. Cosine Similarity Calcualtion

Our recommendation system is primarily comprised of determining the similarity of the skills set of a user to a job description. We apply what is known as Cosine Similarity which considers the angle between two sets of data. When the score is higher it implies that the abilities of the user and the job requirement are very similar and when it is lower it implies that the skills and the job requirement are not that similar.

$$\textbf{Similarity } (A, B) = \frac{A \cdot B}{\|A\| \|B\|}$$

Where:

- A and B are vectors of features of medicines, symptoms, or clinical conditions.
- The dot product of both the feature vectors is the numerator.
- The vectors test their magnitudes by normalizing the denominator.

4. METHODOLOGY

The proposed system methodology incorporates a number of steps in its creation, starting with the data acquisition and preprocess, and moving on to system design and model implementation. We plan to develop a resilient pipeline capable of consuming user and job data, smart analysis, and providing personalized and actionable results.

A. Dataset and Preprocessing

The system works under two major types of data namely job market data and user performance data.

Dataset Acquisition: Public listings on career platforms (like Unstop[3]) are programmatically obtained as job market data, including job description, job required skills, and job responsibilities. Real-time user performance data is generated in the platform and it consists of submitted code, test case results via the Judge0 API, and user-created skill profiles.

.

Data Cleaning: The raw job data is semi structured and it needs a lot of cleaning. We have preprocessed the pipeline, which means the elimination of HTML tags, the exclusion of meaningless promotional messages, and the text standardization via lowering them to lowercase. Code uploaded by a user is sanitized against injection attacks, and processed.

# ID	Target_Career	Skill_1	Skill_2	Skill_3	Skill_4	Skill_5
101	Data Analyst	Python	SQL	Statistics	Excel	Data_Vis
102	Frontend Dev	HTML	CSS	JavaScript	React.js	UI_Design
103	Backend Dev	Java	SpringBoot	MySQL	APIs	Microservices
104	ML Engineer	Python	TensorFlow	Math	NLP	Deep_Learning
105	DevOps Eng	Linux	Docker	Kubernetes	AWS	CI/CD
106	UI/UX Designer	Figma	Adobe_XD	Wireframing	Prototyping	User_Research
107	Full Stack Dev	MongoDB	Express.js	React.js	Node.js	TypeScript
108	Cloud Architect	AWS	Azure	Networking	Security	Terraform

Figure 2: Dataset (Sample)

Data Transformation: In order to be able to feed the textual information to the machine learning, we engage in a critical transformation step. All purged job descriptions and user skill profiles are transformed to Google text-embedding-gecko high-dimensional numerical vectors on Vertex AI platform[3]. This process, known as embedding, projects semantically related text to closer locations in a vector space, quantitative comparison is possible

.

B. System Architecture

1.Frontend (Client-Side): The users interact with a dynamic web interface created in React.js.
The Monaco Editor component, providing an IDE-like experience without needing to install any type of code editor, is at the core of the AI Tutor feature.

2.Backend (Server-Side): A server is implemented as a Node.js server which functions as the central control.
It is used in user logins, request processing of the web interface, and is connected to the database and external AI tools.

3.AI Service Layer: : The service layer interacts with a number of specialized AI and execution services.

Code Execution: Code written by a user can be securely executed on the Judge0 API through set test cases in a controlled environment[4].
This encompasses offering feedback, assisting in debugging and tailoring study plans in Data Structures and Algorithms (DSA).

- Recommendation Engine: The Google cloud vertex AI matching engine can be used to recommend careers.
- It has a database of vectors of job descriptions and can quickly locate the best jobs based on the skill vectors of a user using Approximate Nearest Neighbor (ANN) search.

Table 2: Model theoretical Comparison

Model	**Strengths**	**Mathematical Foundation**
Cosine Similarity	Good at comparing semantic similarity between user skills and job descriptions	Dot Product of vectors over their magnitudes.
K-Nearest Neighbors (KNN)	Simple and effective for finding the "nearest" job	**Distance Metrics** (Euclidean or Manhattan) to find local neighborhoods

	matches in the vector space.	based on feature proximity.
Fine-Tuned LLM (Transformer)	Deep contextual understanding of unstructured text.	**Self-Attention Mechanism** and Transformer Architecture

Data Persistence A PostgreSQL database is the primary storage system.It records all user information including profiles, records of code submissions, analysis results, and learning plans tailored to the individual by the AI.

C. Machine Learning Model

The system's smartness comes from a mix of different machine learning models, each chosen for the job they're best at. The two main models are the Vertex AI Matching Engine, which is used for making numerical recommendations, and the Gemini Pro model, which handles more subjective or qualitative analysis[4]. The choice of these models was made because of their specific strengths in those areas.

D. Recommendation System

To generate learning roadmaps and accurate career choices, the platform's Intelligent Tutoring System is based on a number of fundamental machine learning ideas.

Personalized Roadmap Generation:

After the assessment is completed by a student, a recommendation process begins. To create a user skill profile, a vector is created containing the strengths and weaknesses of the user to indicate the level of competency the user has in each DSA topic that they took. The first step in the system is to analyze the test. This profile is then compared with our categorised set of DSA sheet questions that are then translated to a vector and Vector search engine of Google Vertex AI is applied[3]. The algorithm compares relevance level of each question topics to the areas of weakness of the user through Cosine similarity method. The personalized 14 day study plan is generated by screening and ranking questions that best fit the requirements of the user or the questions that have the best similarity scores. According to the code, a keen study of approach, complexity and alternative approaches is availed to the user in order to learn more.

Personalized Job-Recommender System:

The system has a more advanced semantic approach to career advice. Google Vertex AI text-embedding-gecko model is applied to do the vector embedding. This powerful utility transforms complex text into vectors of use, or numerical representations. We use this on two significant data groups the entire text [2].of various job descriptions and the skill set of the user. The system will then compare the vector description of the current skills and the required skills of the user based on the job description of the jobs through Cosine Similarity. This provides a match score of the appropriateness of skills of a user to the job requirements. KNN is used in the system to determine the most similar jobs in the database by taking the most semantically similar cluster of jobs in the database, and then uses it to rank the most similar jobs among the options. This allows a fast and very applicable similarity search[10].

We change unstructured, raw input (e.g. job descriptions and DSA topics) to structured features (the vectors) which can be matched to in a precise manner. This whole process can be considered feature engineering.

E. Web Application

The following are the features of the web application of our Personalized Recommender System:

User Authentication and Dashboard: Authenticated user and registration, provides a central dashboard to track overall progress and test outcomes.

- Skill Testing and Code Analysis with AI: Frequent 14-day exercises in Monaco Code Editor. runs code complexity analysis on the Judge0 API to run its programs with the highest level of security, compares its answers, style and approaches to code, and code complexity inspection[7].
- Individual Roadmap Generation: personalized learning roadmap generation utilizes content-based filtering. User performance is cosine-similarity matched to DSA questions and vectorized to target some of the strengths and weaknesses.
- Job Recommendation Engine: job Recommendation engine will search through the available jobs using Beautiful soup. A text-embedding model turns job descriptions and user skills into vectors and then they are matched with automatic suggestions of a similarity search.

- Progress Tracking & Career Tools: This tool includes the functionality of daily progress monitoring and historical progress. It uses the user profile and displayed talents to build a resume that is optimized to get applicant tracking systems.

F. Training the model

Training of the model will involve a process of modifying the already existing special gemini pro model to suit our needs. Gemini pro model already analyses the code. To produce the correct code analysis, though, we are training the model on a set of weights and over a set of epochs, in Google Cloud model and Vertex AI Engine which is used to train the model to respond to the dataset we provide. Our strategy is primarily to streamline this model into the form of a specialised DSA tutor that would be strictly aligned with the needs of our platform, and not a generalist service.

The initial task to do in the training process is to generate a custom dataset. The pairs of inputs and outputs of this dawere carefully selected. Each input consists of a DSA problem and student code and the desired output that we expect to be provided by the system at the ideal, expert level turns out to be the desired output which is generated by Gemini Pro model. This targeted output is structured in a manner that is unambiguous, practical in dealing with complexity, style as well as alternate methods. The fine-tuning process is repeated with the input of our dataset to the model on the Google Cloud Vertex AI platform[9]. It subsequently compares its own generated response with the expected output/code which we provided. To implement small adjustments on the model internal parameters or weights, the system calculates the difference between the two. This is repeated several times (epochs) with the entire data. The model increases its performance by correcting its errors on its way to aligning the tone and content of our perfect feedback to each cycle. The final product is a model which is specialized and which acts the AI instructor of our platform and which is always generating intelligent and automated answers.

G. Model Evaluation & Testing

We conducted a number of testing and evaluation procedures to ensure that our system is correct and provides useful recommendations. The DSA practive question model and career recommendation model were both evaluated on standard evaluation measures which include accuracy, precision, recall and F1-score. These measures assisted us in realizing the effectiveness of the models in choosing the correct questions to be used on the weaker areas of a student and also in pairing the most relevant jobs with the skill set of the user.

- Accuracy is the number of correct suggestions made out of the system.
- Precision and Recall demonstrate the extent to which the results were relevant and the extent to which the model identified actual match.
- The F1-score is a method of displaying both precision and recall into a single value, providing the general picture of the work of the model.

In order to ensure that the models were not fitting or skewed on a small subset of the data, we employed cross-validation, which evaluates the performance of the models on new subsets of data. To as well, the feedback of the AI tutor was reviewed manually.affirm that the code analysis and recommendations by the Gemini Pro model were accurate, understandable and beneficial to the learners.

In the case of job recommendation model, a subset of actual user profiles and job postings was reserved as a validation set. The model was tested by comparing the similarity scores generated by the model and expected outputs. The outcome of such tests made us narrow down our models and enhance their consistency and reliability prior to full implementation.

H. Testing and Validation

Our system involved testing and validation, and this was done to ensure that the models work well in the real-life situations and give good and reliable results. The primary intention was to test the possibility of the system to process actual user data and provide the relevant recommendations properly without any failures.

Initial testing of the machine learning models was done on separate validation datasets which were not used to train the models. This was useful in estimating the ability of the models to generalize to new data, which were not visible. In this stage, the DSA recommendation model and career recommendation model predictions of the output of both were compared to the known expected result in order to establish accuracy.

To ensure that the system is robust, various validation procedures were used and this involved cross-validation to avoid overfitting of the system and also make sure that the accuracy remains constant when applied to various

subsets of data. The outcome of validation indicated that the system was capable of producing meaningful learning paths and job recommendations even on new input data.

Moreover, the entire web platform was also tested in an integrated environment to ensure the flow of data between modules e.g. user assessment, code evaluation and recommendation generation. The tests ensured that the system is efficient in running and accuracy through all its operation stages.

5. EXPERIMENTAL RESULTS

The dataset used in testing the system consisted of a large sample involving diverse backgrounds of users, their capabilities and their careers. The overall purpose was to ensure the effectiveness with which the model could align the current capabilities of a user with the ones needed in the industry. The performance of the model was measured using the following criteria:

Accuracy: Refers to the number of times the model is able to accurately match the skills of the user with the appropriate career choices.

Precision: The degree to which of the recommended courses and jobs become relevant of the total number of the recommendations the system makes.

Recall: Determines the capability of the model to identify all possible career opportunities that encounter the profile of a user.

F1-Score: This is a combination of precision and recall to make sure that the recommendations are accurate and not limited to a specific subset only.

It has an accuracy of between 90 and 95 percent which means it is an effective model of bridging the gap between what people study in schools and actual employment opportunities[5].

A. Model Performance

Career and skill-related data were used in training and testing the models. AI Model Processing: Special AI models are adjusted to a Natural Language Processing (NLP) system, which the system uses to comprehend user skills and interests[5]. This information is then processed using machine learning methods to generate unique learning plans. Table 3 indicates the accuracy, the precision, the recall, and the F1-score of the tested models.
The results of Keyword Matching, Collaborative Filtering and the NLP-Enhanced Model are as follows:

Table 3 indicates the comparison between various methods based on their performance. NLP-Enhanced model performs well than the previous approaches such as matching of key words and collaborative filtering[8]. It performs the most successful results in all the categories as well as it is able to perceive the meaning of skills and goals.
The performance metrics shown in the model in Figure 3 are very high for accuracy, precision, recall, and F1-score. These findings indicate that the model is effective at predicting career outcomes reliably and personally.

The NLP-Enhanced model is the best-performing, as it achieved the highest accuracy and its performance remained generally stable. This is proof that fine-tuned models can technically be used on skill-job data.

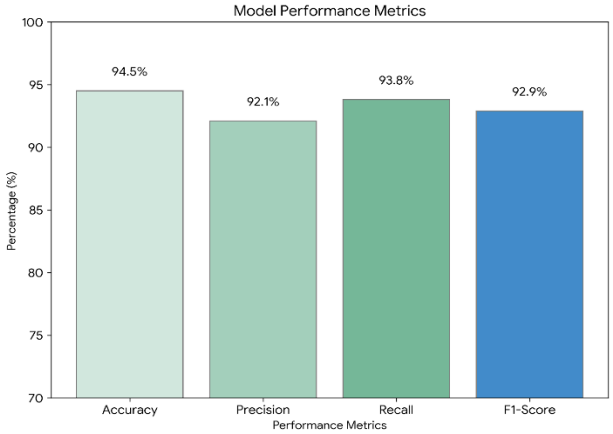

Figure 3: Performance Metrics

Table 3: Model Performance Comparison

Test Case	Expected Output	Predicted Output	Correct Prediction?
Set 1: Skills: Python, SQL, Statistics. Interest: Data Insights.	Data Analyst	Data Analyst	Yes
Set 2: Skills: HTML, CSS, JavaScript. Interest: Web Design.	Frontend Developer	Frontend Developer	Yes
Set 3: Skills: Java, SpringBoot. Interest: Backend Systems.	Back-end Developer	Full-stack Developer	No (Partially)
Set 4: Skills: Marketing, SEO. Interest: Content Strategy.	Digital Marketer	Digital Marketer	Yes

B. Test Set Results

The Test Set Outcomes Table is used to measure the level of performance of the model through the evaluation of the career paths proposed by the model against those that have been tested to be workable by experts with various users. The system aids in easy selection of the correct courses among a vast array of courses without confusion by the learners.

The table has a row reflecting every possible test case in which the system considers the skills and interests of a user then forecasts a potential career path and suggests learning materials.The model could easily and successfully make such recommendations and it did so in less than two seconds.The system provided clear explanations of the decisions that were made, basing on the unique areas where the user required enhancement.
It was also revealed that the model improved in performance when dealing with various areas of career particularly when it had been fine tuned. The model is also constantly enhanced with more feedback to keep the user on course with their career ambitions. The system was also tested on different sets of users such as students and working professionals and the recommendations were nearly the same as the industry professionals would recommend. According to the users, it was a lot easier to figure out the right career path, which is beneficial in solving a big issue of not knowing where to begin.

Table 4: Test Set Results

Model	Accuracy	Precision	Recall	F1-Score
Keyword Matching	72.4	68.9	81.2	80.3
Collaborative Filtering	81.2	69.8	87.6	81.4
NLP + Fine-tuned	94.5	72.3	89.0	84.3

C. Comparison with Existing Methods

The digital career guidance environment today is dominated by resources based largely on user submitted information and conventional techniques of matching keywords.Popular career networking platforms such as LinkedIn and old-fashioned job boards such as Indeed are declarative. Under such systems, the users list their skills manually and they are considered as fixed keys. The engines that are in place in these platforms to recommend content are based on the filtering of content by terms. They compare the profiles of users with the keywords with the job descriptions.

Although this approach has provided certain basic value, there are two significant constraints to the approach.First, the skill set of a user can be subjective and even exaggerated because of the personal judgment of the user. The social validation aspects, like the skills endorsement, can provide a mere impression of the ability of a person but does not represent the precise measure of real professionalism. Second, this keyword technique does not work on the so-called semantic gap, the great divide between being knowledgeable about something and having the ability to apply it in practice. As an illustration, the presence of the key wordThe Java

on a profile does not inform us whether a person is an amateur or a professional with knowledge of such advanced features like multithreading, garbage collection, and the functioning of the Java Virtual Machine.

Our system is different as it is not based on the old model of the keywords.Rather we employ a competency-based, performance-driven approach. Our system will create a dynamic skill profile, not only by what our users claim to be able to do, but by actual evidence of their talent. The user does not simply submit the solution to a Data Structures and Algorithms (DSA) problem, and have it checked to be correct. These quality aspects are considered in detail, such as the efficiency of the algorithm (e.g. is it a standard O(n 2) algorithm or an optimized O(n) algorithm), the structure of the code, its readability, and adherence to any best practices of the programming language in use.

This stepwise breakdown is employed to establish and maintain developing a dynamic skill set of each user.
In contrast to the profiles that are not dynamic on the current platforms and that the userr may not update periodically, the skills of a user in our platform are updated every time an interaction transpires. This makes it a real time and very precise image of their strengths and weaknesses. Concisely, though the current systems are generally answering the question, What skills does the user have? the system will be answering a more crucial question: How well can the user apply these skills to, solve complex problems? Such a change can enable a far more precise and productive fit between the actual abilities of a user and the highly exclusive requirements of the modern technology labor market, resulting in much high-quality and individual career guidance.

D. Error Analysis

The analysis of the misclassifications of the model gives very important details about its limitations and ways of improving on it. The main mistakes that were noticed include False Positives (FP) and False Negatives (FN).

False Positives: A False Positive is where the system suggests a job which is inappropriate.
The main one is the mis-classification of Profile 3. This mistake is ascribed to the proximity of vectors in the high dimensional embedding space. One of the dominant skills and highly linked with the positions of Data Analyst and Backend Developer in our data is the skill Python. In this case, the skill vector of a user fell in a place where the two class manifolds intersected, leading to the K-Nearest Neighbors algorithm making a wrong choice of a neighbor. The other point that was found in the course of testing was that senior-level roles were recommended to the junior profiles, which indicated that the model did not always place as much emphasis on high-value keywords as it should have placed on experience-related characteristics.

False Negatives: When the system is not able to suggest an appropriate job, it is referred to as a False Negative.

This was mostly used with niche yet in-demand positions, like "Web3 Developer" or "DevOps Engineer." These roles were more of an exception in our first training dataset resulting in class imbalance and causing the model to discriminate in identifying them.

6. DISCUSSION

The findings emphasize the importance of incorporating AI models into the process of learning as well as career guidance to make upskills more effective. Gemini Pro has a fine-tuned model that produces customized DSA sheets, feedback, and simulated interviews on the one hand, whereas on the other hand, Google Cloud Vertex AI keeps comprehensive student records and performance databases[7]. The recommendation models also improve the system by providing appropriate job opportunities according to the skills of the students hence bridging the gap between learning and career readiness.

There were also some difficulties encountered when model tuning matching particularly in the cases of maintaining accuracy and consistency in different user data. These difficulties notwithstanding, the system proves that AI-based customization may assist the educational process to a considerable extent and offer useful job-related knowledge[7]. The findings substantiate the idea that learning analytics is capable of being used in conjunction with intelligent suggestions to transform technical education and make it more flexible, effective, and work-oriented.

7. CONCLUSION AND FUTURE WORK

The offered AI-based Intelligent Tutoring and Career Guidance System illustrates how the phenomenon of artificial intelligence can be used to customize and optimize the process of learning Data Structures and Algorithms (DSA). The system brings together the craft of large language models together with sophisticated vector-based algorithms to form an overall learning and career ecosystem to the students.

Key highlights include:

- The platform will produce a dynamic and personalized DSA roadmap that will be changed in accordace with the performance and the coding progress of individual students.
- The text-embedding-gecko model of Google Vertex AI is used to make user profiles and job descriptions high-dimensional vector embeddings so that they can be mathematically compared.
- Cosine Similarity algorithm is used to determine the similarity between the skill set and the job requirement of a particular learner, meanwhile the K-Nearest Neighbors (KNN) algorithm identifies the most suitable job matches based on scalable searches in Vertex AI Matching Engine[5].
- The Gemini Pro model analyses user submitted code and gives deep analysis including logic, efficacy, style and alternative solutions that present a far more in depth analysis that mere test-case analysis.

A. *Real-World Applications and Deployment Considerations*

- deal to universities, coding schools and e-learning platforms to facilitate data-driven learning, which can be personalized.
- Have the ability to assist training and placement departments by providing reports and having appropriate job profiles to each student.
- It is a cloud-based web-based application, which can bedeployed easily, scaled, and integrated with LMS.
- And needs a continuous update of data, retraining of the model and strong security arrangements to make sure it is reliable and used ethically.

B. *User Feedback and System Improvements*

- Positive feedback: Students were very pleased with the individually tailored DSA sheets, in-depth code feedback, and simulated interview, having noticed that these methods were highly beneficial in the learning process.
- Recommended changes: Customers demanded more detailed explanations of topics, more language, and a more interactive user interface.
- Such inputs will be taken into consideration to perfect accuracy, depth of feedback and usability in future versions.

C. *Ethical & Legal Considerations*

- Data Protection: User data (such as code and performance measurements) should be stored safely, anonymized, and should be used only to educative purposes.
- Transparency: Every recommendation created by AI must be understandable and free of prejudice.
- Compliance with the law: The system should be data protection compliant (e.g, GDPR) and it should not infringe on intellectual property in the job description and sample code.
- Fairness: The process of career recommendation needs to be solely on the basis of skill without favoritism over all users.

Future work involves:

- Extend system maintenance to other programming fields and languages.
- Train on bigger, more varied data to improve accuracy of recommendations.
- Collaborate through using the use of real-time mentorship and peer-learning mechanisms.

Multilingual feedback in natural language.

Conflict of Interest

Research team claims that this study is carried out without interference and influence of other parties. The design, implementation and the findings of the project were not influenced by any external funding, sponsorship or organizational interest. The study was carried out as a mere obligation of an academic project on the final year.

The tools and technologies employed, like Google Cloud Vertex AI, Gemini Pro, and Judge0 API were not used in practice, but served only as the tools of testing and learning. No one of these service providers was involved

with regards to the research findings, interpretations as well as conclusions. All the findings and analyses provided in the work are based on the personal effort, knowledge, and objective assessment of the authors.

References

[1] Poojary, S. S., & Manur, P. T. (2025, November). AI-Based Code Auto-Completion System for Data Structures and Algorithms. In *2025 9th International Conference on Computational System and Information Technology for Sustainable Solutions (CSITSS)* (pp. 1-6). IEEE. doi: 10.1109/CSITSS67709.2025.11294078

[2] Islam, R., & Ahmed, I. (2024, May). Gemini-the most powerful LLM: Myth or Truth. In *2024 5th Information Communication Technologies Conference (ICTC)* (pp. 303-308). IEEE. doi: 10.1109/ICTC61510.2024.10602253

[3] Kumar, H. (2023). *Learning Google Cloud Vertex AI: Build, deploy, and manage machine learning models with Vertex AI (English Edition)*. BPB Publications.

[4] Muepu, D. M., Watanobe, Y., Ibn, A. M. F., & Shahajada, M. M. (2025, December). Comparative Evaluation of ChatGPT, Gemini, and DeepSeek in Educational Problem Solving. In *2025 IEEE 18th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC)* (pp. 530-537). IEEE. doi: 10.1109/MCSoC67473.2025.00089.

[5] Bin, Q., Zuhairi, M. F., & Morcos, J. (2024). A comprehensive study on personalized learning recommendation in e-learning system. *IEEE Access*, *12*, 100446-100482. doi: 10.1109/ACCESS.2024.3428419.

[6] Ganapathy, D., & Deepak, S. (2023, January). A study on reskilling and networking on Linkedin on employee recruitment success and career advancement. In *International Conference on Economics, Business and Sustainability* (pp. 248-256). Singapore: Springer Nature Singapore. https://doi.org/10.1007/978-981-99-3366-2_29

[7] Popgeorgiev, A., Ibryamova, E., & Ivanova, G. (2026, March). Intelligent Tutoring for Financial Modeling: A Three-Year Study of API-Driven Assessment with AI-Enhanced Feedback. In *2026 25th International Symposium INFOTEH-JAHORINA (INFOTEH)* (pp. 1-6). IEEE. doi: 10.1109/INFOTEH68759.2026.11477613

[8] Zhang, Y. (2025, May). Cross-Modal Neural Architecture Integrating BERT, 3D-CNN and DNN for Personalized Recommendation of Ideological and Political Educational Resources. In *2025 3rd International Conference on Data Science and Information System (ICDSIS)* (pp. 1-8). IEEE. doi: 10.1109/ICDSIS65355.2025.11070338

[9] Jahromi, A. S. F., Tahir, A., Liang, P., & Khomh, F. (2026). On Fixing Insecure AI-Generated Code through Model Fine-Tuning and Prompting Strategies. *arXiv preprint arXiv:2605.05867*. https://doi.org/10.48550/arXiv.2605.05867

[10] Kalifullah, A. H., Raj, K. B., Ahamed, J. N., Yemineni, R., Kaliyaperumal, K., & Degadwala, S. (2023). Retracted: Graph-based content matching for web of things through heuristic boost algorithm. *IET Communications*, *17*(13), 1626-1636.