XCRS: an Explainable Course Recommendation System for Information Technology Careers Powered by LLMs

Muhammed Yasin Horasanlı
Boğaziçi University
Istanbul, Turkey *muhammed.horasanli@std.bogazici.edu.tr*

Fatma Başak Aydemir
Utrecht University
Utrecht, The Netherlands *f.b.aydemir@uu.nl*

Hüseyin Birkan Yilmaz
Boğaziçi University
Istanbul, Turkey *birkan.yilmaz@bogazici.edu.tr*

Abstract

The growing number of online resources on information technology has left many learners feeling overwhelmed by the large number of career options and the paths to achieve them. This abundance of choices highlights the need for personalized career guidance and clear course recommendations to help learners focus on their specific goals. Existing recommendation systems fail to provide transparency and clear explanations for their suggestions. To bridge this gap, we present XCRS: Explainable Course Recommendation System, which recommends both career roles and associated courses in information technology with explainability at its core. XCRS utilizes large language model embeddings from Google, OpenAI, MistralAI, VoyageAI, and Cohere to deliver personalized recommendations tailored to users' knowledge, past preferences, and future learning interests. Our contributions are two-fold: *i*) a pipeline to construct an explainable recommendation system for career pathways in information technology, *ii*) a replication package that includes the implementation, a public dataset of information technology courses, and the design for empirical evaluation. Our evaluation suggests that the overall system has been perceived as useful by the intended users, while there is no statistically significant difference in the performance of the large language models used.

Keywords: Course Recommendation System, LLMs, Vector Similarity Search, Explainability.

1. Introduction

Recommendation systems are algorithms designed to suggest relevant items based on user data and preferences, influencing decision-making and enhancing user experiences across industries like e-commerce, streaming, and healthcare [13]. By identifying patterns in user behavior, these systems personalize content and product suggestions, improving customer satisfaction and retention. In the education sector, the demand for personalized recommendation systems has increased, driven by the rise of Massive Open Online Courses (MOOCs), which served more than 220 million learners worldwide in 2021 [20]. With a market valued at $20.53 billion in 2023 [15], platforms like Udemy [23], edX [8], and Coursera [7] exemplify the growing need for accessible and flexible education. However, while these platforms offer course recommendations, their decision-making processes often lack transparency, leaving users uncertain about the recommendations.

Given the increase in demand for information technology (IT) related skills and the increasing number of online courses available for learning these skills, there is a pressing need for a recommendation system that not only suggests relevant courses but also explains its reasoning.

Traditional university students find it difficult to choose a specialization or plan their career path due to the variety of emerging roles in the IT sector [24]. Similarly, self-taught learners who wish to shift their careers into information technology often struggle to identify the right courses or career roles that match their goals and interests. With the plethora of courses available, the need for guidance that goes beyond 'black-box' recommendations becomes even more apparent.

We introduce the Explainable Course Recommendation System (XCRS)[1], a novel system designed to provide both personalized and explainable recommendations for roles and associated courses within the IT domain. XCRS is built to achieve three main objectives: generating career role and course recommendations aligned with user's unique profiles, providing clear explanations that help users make informed decisions about their learning journeys, and improving the recommendation process by integrating multiple large language models (LLMs) to capture diverse perspectives and offer varied options.

Firstly, we scraped a dataset from the MOOC platform Udemy [23], focusing on IT courses. This dataset forms the foundation for tailoring suggestions to users' background knowledge and areas of interest. We designed the system with a user-friendly interface that facilitates seamless navigation and provides clear explanations to build user trust and transparency. Additionally, we integrated multiple LLMs into the system to generate top-k recommendations based on user inputs to compare results across different approaches. Finally, we evaluated the system's performance through a user study involving participants from computer science and IT-related backgrounds, assessing dimensions such as effectiveness, persuasiveness, transparency, efficiency, serendipity, and satisfaction. These efforts resulted in the following key contributions, and the remainder of the paper is structured to support these:

- We introduce a pipeline to construct an explainable recommendation system for career pathways in information technology.
- We make available a replication package[2] that includes the implementation, public dataset of information technology courses, and design for empirical evaluation.

2. Related Work

To position our contribution within the existing literature, we discuss prior work on explainable recommendation systems, course and career recommendation approaches, and the use of large language models in this context.

2.1. Explainable Recommender Systems

Explainable recommendation systems have gained traction as a way to improve user trust and engagement by offering clear justifications for recommendations. Approaches such as Wang et al.'s multi-task learning model [26], Chen et al.'s Neural Attentional Regression model (NARRE) [4], and reinforcement learning-based frameworks [27] have demonstrated the effectiveness of combining recommendation systems with different types of explainable mechanisms. These methods have been evaluated in diverse domains, from e-commerce [5] to education [21], [10], demonstrating the ability of explanations to improve user satisfaction and decision-making. While purposes may differ across domains, Tintarev and Masthoff [22] identify seven potential goals for explanations in recommendation systems: transparency, scrutability, trust, effectiveness, persuasiveness, efficiency, and satisfaction. In practice, generating explanations that excel in all these goals is challenging, as it often involves balancing trade-offs [22]. Therefore, we focused on achieving specific goals by incorporating explanations into recommendations using LLMs.

[1] Demo: `https://xcrs.vercel.app` (home page only; recommendation feature unavailable).
[2] `https://doi.org/10.5281/zenodo.14291087`

2.2. Recommender Systems for Career and Education

In education and career guidance, recommendation systems have evolved from content-based [17] and collaborative filtering methods [9] to deep learning [14], [19] and knowledge-based approaches [12]. Recent work has also explored aligning course and career recommendations with evolving job market needs and user preferences. For instance, Striebel et al. [21] proposed a knowledge graph-based course recommender that generates explainable recommendations for predefined career paths. Their approach depends on structured ontologies and assumes explicit career goals from users. Our work instead uses user self-assessment and open-ended inputs, offering a more flexible framework that adapts to diverse learning backgrounds and evolving interests. Frej et al. [10] introduced a course recommendation system that incorporates job market trends through skill extraction using LLMs and generates sequential course paths using reinforcement learning. While their work emphasizes market alignment and minimal supervision, our system differs in its focus on role-oriented guidance through the roadmap.sh [1] data, model diversity through multiple LLM embeddings, and transparent, comparative recommendation outputs. Beutling and Spahic-Bogdanovic [2] developed a course recommender that links predefined job roles to course learning objectives using an ontology and ChatGPT. Their system focuses on a single academic program and relies on fixed course descriptions and a structured ontology to generate recommendations. In contrast, our system supports open-ended inputs, recommends both roles and courses, and generates tailored explanations with LLMs. We recommend both courses and career roles within an interface that allows users to compare results across LLMs. Our approach fills a gap in the literature by integrating personalized user feedback, community-driven role structures, and multi-model embeddings into an explainable system that supports both course and career guidance, offering a reproducible pipeline grounded in real-world learning contexts.

2.3. Large Language Models

The integration of LLMs into recommendation systems has transformed the field, enabling systems to generalize, personalize, and address complex tasks more effectively. Surveys by Wu et al. [29], Zhao et al. [30], and Chen et al. [3] categorize LLMs into paradigms such as feature extractors, autonomous recommenders, and conversational agents, highlighting their versatility and increasing adoption in both academic and industrial applications. These studies emphasize the potential of LLMs to enhance recommendation accuracy and provide personalized, context-aware suggestions. Accordingly, we employed five different LLMs to generate embeddings that support the recommendation process.

3. Approach

The research goal is to provide personalized and transparent recommendations, ensuring that the recommendations are not only accurate but also explainable. Figure 1 presents the steps taken to implement our approach.

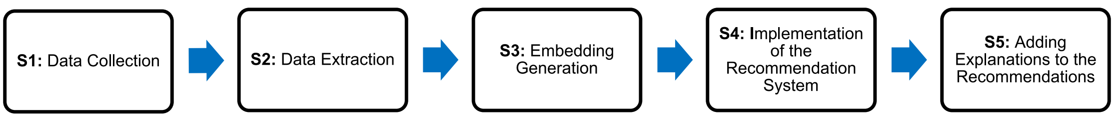

Fig. 1. Summary of the steps in the implementation process of the XCRS.

In S1, the method begins with data collection, capturing course-related information from Udemy and IT career roadmaps from roadmap.sh project. In S2, the collected data are processed and organized into a structured dataset. In S3, embeddings are generated to create representations of the collected data and user inputs using state-of-the-art LLMs. In S4, the implementation of the recommendation system involves matching user knowledge with roles and courses, constructing similarity matrices, and generating recommendations based on the user's familiarity with roadmap concepts and learning goals. In S5, we integrate explainability into the recommendations, providing users with contextually relevant explanations generated by LLMs.

3.1. Data Collection

The initial step in building the recommendation system (S1) involved collecting a comprehensive dataset of online courses and career roadmaps. To collect course data, we utilized Udemy's Affiliate API, which provided essential metadata for each course, such as title, URL, headline, language, and price. To supplement this, we employed web scraping techniques to extract additional fields like course descriptions, relevant course categories, and learning objectives directly from the course web pages. In addition to course data, we formed a dataset with IT career roadmaps obtained from the roadmap.sh project. We collected roadmaps for ten popular IT roles: AI-Data Scientist, Android Developer, Backend Developer, Blockchain Developer, Frontend Developer, Full-Stack Developer, DevOps Engineer, Game Developer, Quality Assurance (QA) Engineer, and User Experience (UX) Designer. Each roadmap outlines a structured progression of knowledge and skills necessary for success in the respective role.

3.2. Data Extraction

We focused on extracting specific features to construct a comprehensive representation of each course: title, headline, category, learning objectives (what you will learn), and course description as illustrated in Figure 2 for an example course.

Title	HTML5 and CSS3 Fundamentals
Headline	Build your very own website with HTML5 from scratch using HTML5 and CSS3 - designed for complete beginners
Category	Development, Web Development, CSS
What You'll Learn	Know how to use Html tags and build with the most common ones. Learn the use of attributes and common settings. Create CSS stylesheets that control your site design and set them up on your project site. Understand how the elements go together to build each part of the site Finish a complete typical website as part of your lessons.
Description	When it comes to the world of technology, staying ahead of the curve is always a challenge. In the last year one aspect of this – the world wide web – has kicked up a gear with the introduction of HTML5, the newest version of the code that makes the web tick. If you learn HTML5 along with CSS3 (the next level of web design used on all modern websites), you'll have a recipe for success; and this course will show you how. Create a website from scratch with HMTL5 and CSS3 … …

Fig. 2. Structure of the extracted course data for "HTML5 and CSS3 Fundamentals".

To refine the dataset for our recommendation system, we applied several cleansing steps. Starting with an initial pool of 8,180 courses scraped from Udemy, only English-language courses were retained. We narrowed the scope to relevant categories, specifically Development, IT & Software, and Office Productivity, aligning with targeted roles and user profiles. Duplicate courses were removed, and courses lacking essential information, such as titles, categories, or descriptions, were excluded to ensure data integrity and completeness. After these steps, 7,727 courses were eliminated, resulting in a refined dataset of 453 courses.

To structure the roadmap data, nodes were categorized as either *topics* (which may contain child nodes) or *concepts* (which do not contain child nodes). This classification enabled a hierarchical organization of roles, topics, and concepts, resulting in 235 topics and 869 concepts. Each node was stored with its name, description, and type; however, only concepts will be used in the later stages. These concepts serve as a structured basis for representing user input and courses, effectively functioning as a form of normalization.

3.3. Embedding Generation

Our recommendation system utilizes five distinguished LLMs to generate embeddings, selected based on their performance on the Massive Text Embedding Benchmark (MTEB). We chose Google's [11] *text-embedding-004*, VoyageAI's [25] *voyage-large-2*, OpenAI's [18] *text-embedding-3-large*, MistralAI's [16] *mistral-embed*, and Cohere's [6] *embed-english-v3.0* for their state-of-the-art ability to capture the semantics of textual data.

For courses, we concatenated their title, category, learning objectives, and description from Udemy. Concepts were represented using their name and description from roadmap.sh. Embeddings were pre-generated using all five models and stored separately as shown in Figure 3(a).

The user input processing stage customizes recommendations based on individual experiences and interests. Users freely enter "knowledge units" (e.g., course titles, concepts, technologies, skills), "object-oriented programming", "RESTful APIs", which are embedded using five LLM models as shown in Figure 3(a). To guide inputs, a cheat sheet listing existing concepts and courses is provided. Users also categorize each knowledge unit as Enjoyed (positive learning experiences), Neutral (indifferent experiences), Didn't Enjoy (negative experiences), or Curious About (future learning interests). This categorization helps the system model both prior knowledge and future goals, avoids undesired content, and promotes relevant suggestions.

3.4. Vector Similarity Search and Similarity Matrices

To generate recommendations for career roles and courses, the proposed system utilizes vector similarity search with LLM embeddings. This approach measures the similarity between embeddings to identify connections between user-provided knowledge units and potential recommendations. Among the available similarity metrics, cosine similarity was selected due to its effectiveness in handling high-dimensional, sparse data.

After generating embeddings for courses, concepts, and user input, the system computes cosine similarity scores to construct two matrices: *Course* × *Concept* and *User* × *Concept*, as shown in Figure 3(b). These matrices represent the similarity between course and concept embeddings, and between user input and concepts. They serve as the foundation for generating career role and course recommendations, as detailed in the following sections.

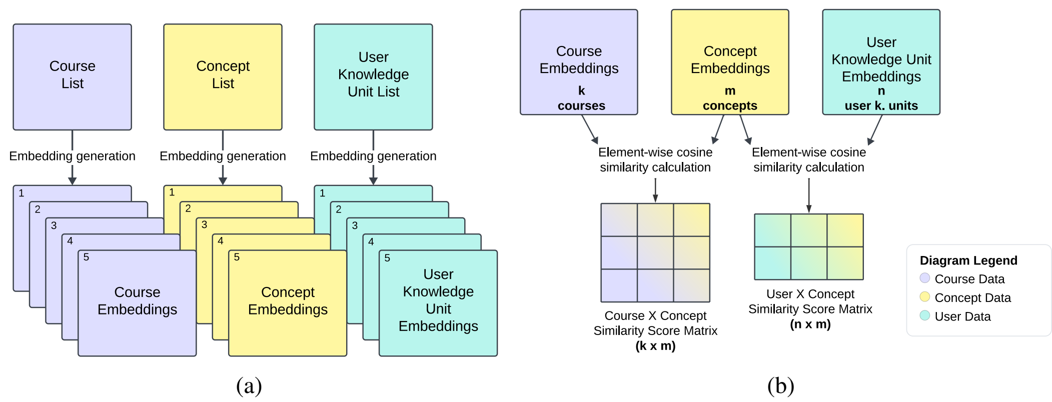

Fig. 3. Overview of embedding generation for courses, concepts, and user knowledge units using five LLMs (a), and similarity matrix creation (b)

Due to variation in similarity scores across LLMs, we analyzed their distributions and found that they follow a Gaussian pattern. After testing several thresholds, we selected the mean plus 2.5 standard deviations, which captures approximately the top 1% of scores for each model.

3.5. Role Recommendation

The role recommendation process uses the *User* × *Concept* similarity matrix to assess how well a user's knowledge aligns with the concepts associated with each career role. Based on this alignment, roles are ranked, and up to three are selected for recommendation, as illustrated in Figure 4.

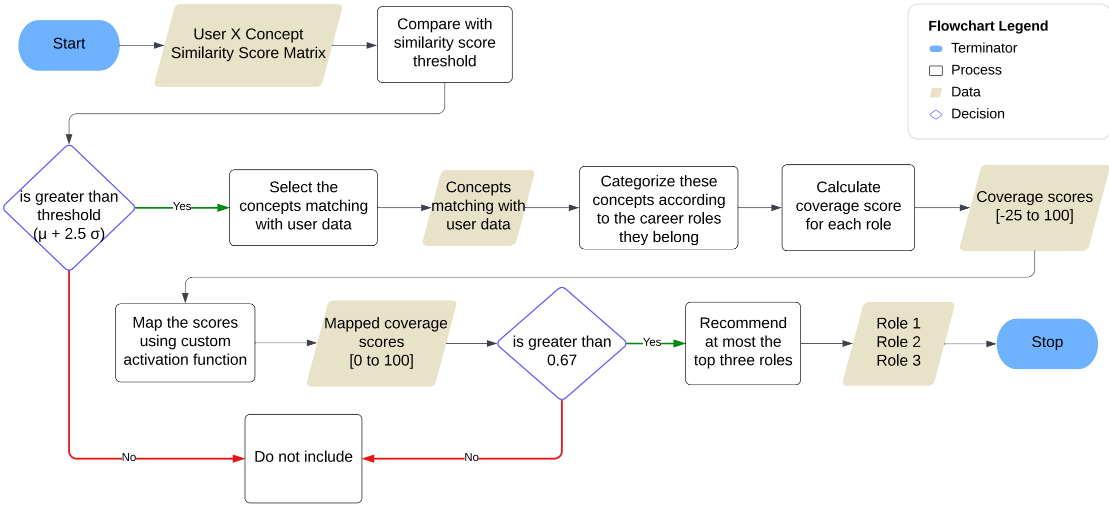

Fig. 4. Detailed process flow for the role recommendation.

To quantify alignment, we employed a role coverage score that reflects the user's engagement with relevant concepts. Categories such as *Enjoyed*, *Neutral*, *Didn't Enjoy*, and *Curious About* are assigned coefficients of 0.75, 0.5, -0.25, and 1, respectively. The weights for each category were determined through a combination of ad hoc reasoning and iterative trials during system development. For each role, the score $S(R_i)$ is calculated by summing the coefficients for all relevant concepts and dividing by the total number of concepts in the role:

$$S(R_i) = \frac{\sum_{j=1}^{n_i} \text{Coef}(C_j^{R_i})}{n_i} \tag{1}$$

where $C_j^{R_i}$ is concept associated with the role R_i, and $\text{Coef}(C_j^{R_i})$ is the coefficient assigned to $C_j^{R_i}$ based on the user's interaction category, and n_i is the total number of concepts in R_i.

To standardize this score between 0 and 100, a custom activation function based on a modified sigmoid is applied, defined as:

$$f(x) = \text{round}\left(\frac{100}{1 + \exp(-0.2 \cdot (x - 25))}, 2\right). \tag{2}$$

The activation function maps the raw score to a 0–100 scale, with higher values indicating stronger alignment. Roles with a score above 0.67 are considered relevant and selected for recommendation, while lower scores suggest insufficient alignment. The top-ranked roles are then passed to the explanation generation component.

3.6. Course Recommendation

The course recommendation process suggests courses based on the three roles identified by the role recommendation module. As shown in Figure 5, which illustrates the flow for one role, the system starts with the user's concept data and assesses their familiarity with the roadmap concepts for that role. A concept is considered familiar to the user when it is found to be similar to a knowledge unit in the *Enjoyed*, *Neutral*, or *Didn't Enjoy* categories, which reflect prior exposure or mastery regardless of preference. Users familiar with over 30% of a role's concepts are classified as intermediate or advanced and are recommended more advanced content. Otherwise, earlier-stage concepts are selected. Familiar concepts are excluded from recommendations.

For each target concept, the system retrieves the top three courses using the *Course* × *Concept* similarity matrix, which ranks courses by their relevance to each concept. Courses corresponding to each concept are aggregated per role. Then, the system ranks the recommendations based on selection count and similarity score, selecting the top three most relevant courses. Finally, these course recommendations are then sent to the explanation generation component.

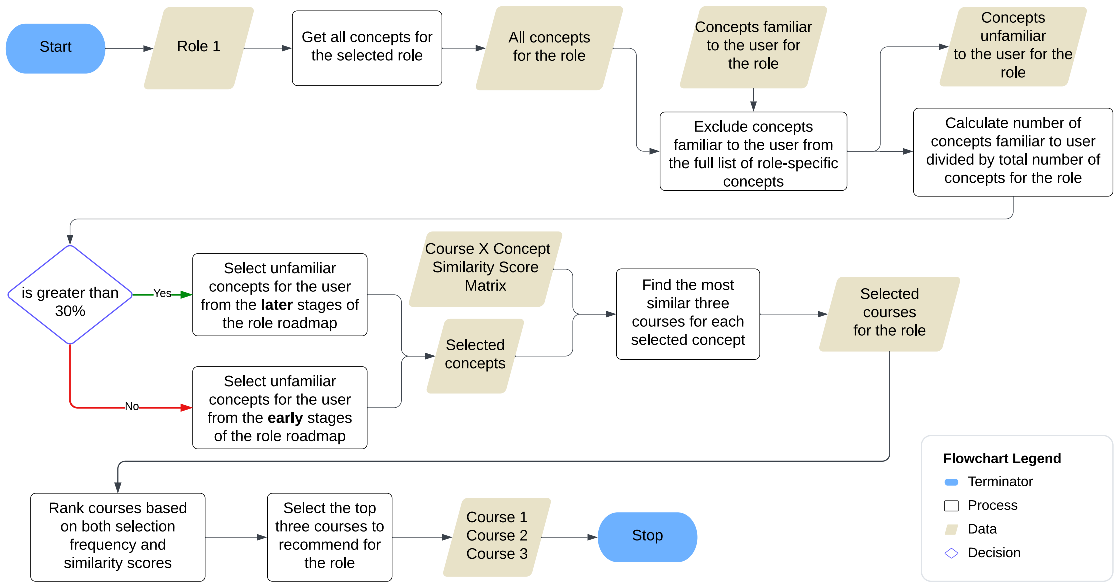

Fig. 5. Detailed process flow for the course recommendation.

3.7. Explanation Generation

The career role and course explanation generation processes aim to provide clear, personalized justifications by linking recommendations to user's knowledge and learning goals. For role explanations, the system considers the user's familiarity with concepts. Explanations emphasize either the user's broad or specific knowledge, beginning with phrases like "I assume you are familiar with" or "I see that you are willing to learn" based on the user's engagement with each concept. Using the GPT-4o model from ChatGPT, these explanations are refined to ensure clarity, expand abbreviations, and make the language more user-friendly, as shown in Figure 6.

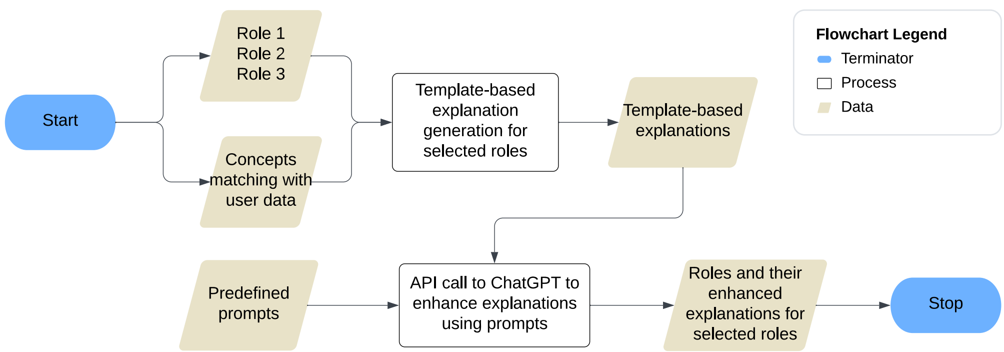

Fig. 6. Flow of explanation generation for the career role recommendation.

The system provides course explanations based on data related to the user's recommended roles, emphasizing concepts the user has not yet mastered. Course content (title, description, learning outcomes) is combined into a prompt highlighting the role name and relevant concepts. This allows ChatGPT to generate explanations based on the user's career roles and existing knowledge gaps by considering course materials.

4. Evaluation

In the user study, we evaluated XCRS from end-users' perspectives, focusing on its performance across several critical dimensions. We aimed to determine how effectively the system meets its target audience's needs.

Participants. We recruited 25 participants from the IT sector, including students and professionals in fields like computer science, electrical engineering, and industrial engineering, providing relevant expertise for evaluating the system.

Procedure. Participants interacted with XCRS by entering knowledge units that they had either mastered or wanted to explore. Based on this input, the system generated personalized career and course recommendations using five anonymized models. Participants evaluated the system through a post-task survey.

Survey Structure. The survey included demographic questions followed by three main sections: usefulness, model satisfaction, and system output evaluation. *Usefulness,* participants rated the usefulness of recommendations for each model. *Model satisfaction,* they identified which model's recommendations they found most satisfying, indicating their personal preference among the five models. *Evaluation of system outputs,* this part included six dimensions:

- Effectiveness (Q1: How satisfied are you with the relevance of the role and course recommendations?),
- Persuasiveness (Q2: How likely are you to consider a course that was recommended to you?),
- Transparency (Q3: How easy do you find the explanations provided for the recommendations to understand?),
- Efficiency (Q4: How easy is it to navigate and use the system to get the recommendations you need?),
- Serendipity (Q5: Do you feel that the system helps you explore courses outside of your usual preferences?).
- Satisfaction - Likelihood to Use: (Q6: How likely are you to use the system to help with your career plan?).
- Satisfaction - Likelihood to Recommend: (Q7: How likely are you to recommend the system to your peers?).

Results. For usefulness, results showed that mistral-embed received the highest average rating for course recommendations at 74.4%, while voyage-large-2 led in role recommendations with a 72.8% usefulness rating. For model satisfaction, participants showed varying preferences: 8 participants favored text-embedding-004, followed by mistral-embed with 6 votes, and embed-english-v3.0 chosen by 5 participants. These results, as shown in Table 1, indicate diverse user preferences, with no single model overwhelmingly preferred in both usefulness and satisfaction.

Table 1. Average ratings and percentage values for the usefulness of recommendations provided by each model. The highest scores are highlighted, and the second-highest scores are underlined.

LLM	Career Roles (Avg. Rating / %)	Courses (Avg. Rating / %)
text-embedding-004 (Google)	3.52 / 70.4%	3.40 / 68%
voyage-large-2 (VoyageAI)	3.64 / 72.8%	3.32 / 66.4%
text-embedding-3-large (OpenAI)	3.32 / 66.4%	3.40 / 68%
mistral-embed (MistralAI)	<u>3.60 / 72%</u>	3.72 / 74.4%
embed-english-v3.0 (Cohere)	3.56 / 71.2%	<u>3.52 / 70.4%</u>

Participants provided detailed feedback on the system's outputs, summarized in Table 2. The system's effectiveness was rated at 80%, with many participants finding the role and course recommendations relevant. Persuasiveness scored 76%, indicating a strong likelihood of considering recommended courses. Transparency received the highest rating at 89.6%, reflecting the clarity and ease of understanding of the explanations. Efficiency also scored highly at 88.8%, with users finding the system intuitive and easy to use. Serendipity was rated at 84%, showing the system's success in introducing new options to users. Satisfaction metrics revealed moderate scores for likelihood to use (69.6%) and recommend the system (77.6%), with most participants expressing positive feedback, though some indicated areas for improvement.

Table 2. Summary of participant feedback on system outputs. Scores above 80% are highlighted.

Dimension	Rating					Std. Dev.	Mean	Mean (%)
	1	2	3	4	5			
Effectiness (Q1)	0	1	6	10	8	0.85	4	80%
Persuasiveness (Q2)	1	1	5	13	5	0.94	3.8	76%
Transparency (Q3)	0	0	0	13	12	0.50	4.48	89.6%
Efficiency (Q4)	1	0	1	8	15	0.90	4.44	88.8%
Serendipity (Q5)	4:No, 21:Yes					0.37	4.2	84%
Satisfaction (Likelihood to Use, Q6)	2	4	3	12	4	1.17	3.48	69.6%
Satisfaction (Likelihood to Recommend, Q7)	1	4	3	7	10	1.18	3.88	77.6%

Statistical Analysis. Non-parametric tests were used to analyze survey responses because the Shapiro-Wilk test showed that the data was not obeying normal distribution. Kruskal-Wallis H Test found significant differences in how persuasive (Q2) the recommendations were, based on education level ($p = 0.0234$) and job experience ($p = 0.0128$). Other dimensions, including effectiveness, transparency, and efficiency, showed consistent responses across groups, suggesting general robustness in system usability across educational and employment backgrounds. Friedman Test is applied to assess ratings across system dimensions (effectiveness, satisfaction, persuasiveness, transparency, efficiency, and serendipity), the Friedman test indicated significant variability ($p < 0.001$), highlighting differences in how participants evaluated these aspects.

The Friedman test was conducted to evaluate the significance on the usefulness comparison among the models. The results indicate that there are no significant differences between the models for either career role ($p = 0.8415$) or course recommendations ($p = 0.6499$). This suggests that all models perform similarly with respect to both recommendations.

5. Discussion

The XCRS demonstrates strengths in transparency and usability, with high ratings for transparency (89.6%) and efficiency (88.8%) in the user study. These results reflect the system's ability to provide clear explanations and an intuitive interface, contributing to positive user engagement. The analysis shows that education and job experience affect how persuasive users find the system. The Kruskal-Wallis H test showed significance in persuasiveness by education level and job experience, while the Friedman test highlighted variability across dimensions like satisfaction and persuasiveness. These findings suggest opportunities to enhance certain aspects of the system to improve overall user engagement.

Although we could not detect a significant difference between the models, the models found most useful by users in career role recommendations were VoyageAI's voyage-large-2 and MistralAI's mistral-embed, respectively, while MistralAI's mistral-embed and Cohere's embed-english-v3.0 were found the most useful in terms of course recommendations. In overall recommendations, users reported that they received the most satisfactory results from Google's text-embedding-004, MistralAI's mistral-embed model, and Cohere's embed-english-v3.0 models, respectively. Therefore, model selection for the proposed system can be based on other factors, such as size, budget, or energy efficiency.

Limitations include the system's focus on the IT domain, reliance on English content, and use of community-contributed roadmap data, which may vary in accuracy. Future enhancements could broaden applicability to other fields, integrate more rigorously validated data sources, and consider a wider array of LLMs to further align with diverse user needs.

We consider potential threats to validity based on the definitions provided in [28].

Internal Validity. Participant familiarity with recommendation systems could influence their perceptions of XCRS, introducing potential biases. To mitigate this, a consistent protocol was used across sessions, ensuring that participants had equal exposure to system information. Additionally, although the study was conducted in English, all participants had advanced English proficiency, which reduced comprehension issues.

External Validity. The user base was largely limited to IT professionals and students, which may affect the generalizability of results to other fields. Including a broader range of fields in future studies could offer a more comprehensive view of XCRS's applicability. Additionally, reliance on community-contributed roadmap data may impact recommendation accuracy; however, the popularity and credibility of this data source partially offset this risk.

Construct Validity. Evaluation metrics were chosen to reflect core aspects of user experience, though they may not capture all influences on decision-making. To ensure construct validity, survey questions were written in clear language based on prior work in recommender system evaluation [22] and pilot-tested with three computer engineering students for clarity.

Conclusion Validity. A relatively small participant pool limits statistical power, affecting the detection of nuanced differences. While non-parametric tests (Kruskal-Wallis H and Friedman) were appropriate, expanding the participant pool could improve sensitivity to subtle variations. Transparent data analysis and cautious interpretation support the reliability of the conclusions.

6. Conclusions

This paper introduces XCRS, an explainable recommendation system tailored to the IT sector, using LLM embeddings to provide personalized and transparent career role and course suggestions. By integrating course data from Udemy and structured roadmaps from roadmap.sh, XCRS delivers recommendations aligned with users' backgrounds and learning goals.

The proposed system is evaluated through a user study, highlighting its strengths in perceived transparency (89.6%), efficiency (88.8%), and effectiveness (80%). The study also demonstrated the potential of XCRS to broaden user perspectives with an 84% serendipity rating. These results underscore XCRS's ability to bridge the gap between recommendation systems and explainable AI, fostering trust and engagement in educational and career guidance.

The system needs to be evaluated further with larger user groups to confirm our findings. Future work could expand the system's applicability to other domains, integrate additional user-specific features, and refine explainability techniques to enhance user trust and engagement.

References

[1] Ahmed, K.: Developer roadmaps, `https://roadmap.sh/`, accessed on September 25, 2024.

[2] Beutling, N., Spahic-Bogdanovic, M.: Personalised course recommender: Linking learning objectives and career goals through competencies. In: Proceedings of the AAAI Symposium Series. pp. 72–81. Stanford, California, USA (2024)

[3] Chen, B., Dai, X., Guo, H., Guo, W., Liu, W., Liu, Y., Qin, J., Tang, R., Wang, Y., Wu, C., et al.: All roads lead to rome: Unveiling the trajectory of recommender systems across the llm era, arXiv:2407.10081 [cs], 2024

[4] Chen, C., Zhang, M., Liu, Y., Ma, S.: Neural attentional rating regression with review-level explanations. In: Proceedings of the World Wide Web Conference. p. 1583–1592. Lyon, France (2018)

[5] Chen, X., Zhang, Y., Xu, H., Cao, Y., Qin, Z., Zha, H.: Visually explainable recommendation, arXiv:1801.10288 [cs], 2018

[6] Cohere: Cohere | the leading ai platform for enterprise, `https://cohere.com/`, accessed on October 23, 2024.

[7] Coursera: Online courses & credentials from top educators, `https://www.coursera.org/`, accessed on September 25, 2024.

[8] Edx: Free online courses by harvard, mit, & more, `https://www.edx.org/`, accessed on September 25, 2024.

[9] Elbadrawy, A., Karypis, G.: Domain-aware grade prediction and top-n course recommendation. In: Proceedings of the 10th ACM Conference on Recommender Systems. pp. 183–190. Boston, MA, USA (2016)

[10] Frej, J., Dai, A., Montariol, S., Bosselut, A., Käser, T.: Course recommender systems need to consider the job market. In: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 522–532. Washington, DC, USA (2024)

[11] Google: Google generative ai overview, `https://ai.google/discover/generativeai/`, accessed on October 23, 2024.

[12] Ibrahim, M.E., Yang, Y., Ndzi, D.L., Yang, G., Al-Maliki, M.: Ontology-based personalized course recommendation framework. IEEE Access 7, pp. 5180–5199 (2018)

[13] Isinkaye, F.O., Folajimi, Y.O., Ojokoh, B.A.: Recommendation systems: Principles, methods and evaluation. Egyptian Informatics Journal 16(3), pp. 261–273 (2015)

[14] Jiang, W., Pardos, Z.A., Wei, Q.: Goal-based course recommendation. In: Proceedings of the 99th international conference on learning analytics & knowledge. p. 36–45 (2019)

[15] MaximizeMarket: Mooc market – global industry analysis and forecast (2024-2030), https://www.maximizemarketresearch.com/market-report/global-mooc-market/22155/, accessed on September 25, 2024.

[16] MistralAI: Mistral ai | frontier ai in your hands, https://mistral.ai/, accessed on October 23, 2024.

[17] Morsomme, R., Alferez, S.V.: Content-based course recommender system for liberal arts education. International Educational Data Mining Society (2019)

[18] OpenAI: Openai, https://openai.com/, accessed on October 23, 2024.

[19] Pardos, Z.A., Fan, Z., Jiang, W.: Connectionist recommendation in the wild: on the utility and scrutability of neural networks for personalized course guidance. User modeling and user-adapted interaction 29(2), pp. 487–525 (2019)

[20] Shah, D.: By the numbers: Moocs in 2021, https://www.classcentral.com/report/mooc-stats-2021/, accessed on September 25, 2024.

[21] Striebel, J., Myers, R., Liu, X.: Career-based explainable course recommendation. In: Sserwanga, I., Goulding, A., Moulaison-Sandy, H., Du, J.T., Soares, A.L., Hessami, V., Frank, R.D. (eds.) Information for a Better World: Normality, Virtuality, Physicality, Inclusivity: 18th International Conference, Virtual Event. pp. 371–382 (2023)

[22] Tintarev, N., Masthoff, J.: A survey of explanations in recommender systems. In: International Conference on Data Engineering Workshop. pp. 801–810. Istanbul, Turkey (2007)

[23] Udemy: Online courses - learn anything, on your schedule, https://www.udemy.com/, accessed on September 25, 2024.

[24] Vo, N.N., Vu, N.H., Vu, T.A., Vu, Q.T., Mach, B.D.: Crs: a hybrid course recommendation system for software engineering education. In: Proceedings of the ACM/IEEE 44th International Conference on Software Engineering: Software Engineering Education and Training. pp. 62–68. Pittsburgh, PA, USA (2022)

[25] VoyageAI: Voyageai: Supercharging search and retrieval for unstructured data, https://www.voyageai.com/, accessed on October 23, 2024.

[26] Wang, N., Wang, H., Jia, Y., Yin, Y.: Explainable recommendation via multi-task learning in opinionated text data. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. p. 165–174. Ann Arbor, MI, USA (2018)

[27] Wang, X., Chen, Y., Yang, J., Wu, L., Wu, Z., Xie, X.: A reinforcement learning framework for explainable recommendation. In: 2018 IEEE International Conference on Data Mining (ICDM). pp. 587–596. Singapore, Singapore (2018)

[28] Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A.: Experimentation in software engineering, vol. 236. Springer, Berlin, Heidelberg (2012)

[29] Wu, L., Zheng, Z., Qiu, Z., Wang, H., Gu, H., Shen, T., Qin, C., Zhu, C., Zhu, H., Liu, Q., et al.: A survey on large language models for recommendation. World Wide Web 27(60) (2024)

[30] Zhao, Z., Fan, W., Li, J., Liu, Y., Mei, X., Wang, Y., Wen, Z., Wang, F., Zhao, X., Tang, J., Li, Q.: Recommender systems in the era of large language models (llms). IEEE Transactions on Knowledge and Data Engineering 36(11), pp. 6889–6907 (2024)