University Course Recommendation Using Large Language Models

Kai-Wen Hee1, a), Choo-Yee Ting1,2, b), Hui-Ngo Goh1, 3, c) and Albert Quek1, 4, d)

1Faculty of Computing and Informatics, Multimedia University, 63100 Cyberjaya, Selangor, Malaysia

2Centre for Big Data and Blockchain Technologies, CoE for Advanced Cloud, Multimedia University, 63100 Cyberjaya, Selangor, Malaysia

3Centre for Natural Language Processing, CoE for Artificial Intelligence, Multimedia University, 63100 Cyberjaya, Selangor, Malaysia

4Centre for Image and Vision Computing, CoE for Artificial Intelligence, Multimedia University, 63100 Cyberjaya, Selangor, Malaysia

b) Corresponding author: cyting@mmu.edu.my

a) 1211100744@student.mmu.edu.my  
c) hngoh@mmu.edu.my

d) quek.albert@mmu.edu.my

**Abstract.** A course recommendation system guides students toward academic programs that fit their preferences and academic achievement. However, traditional rule-based recommendation systems have notable limitations, especially as the demand for personalized education grows. The current systems often rely on pre-trained machine learning algorithms, which struggle with personalization, explainability and integration of unstructured data. By incorporating large language models (LLMs) into the recommendations framework, this study aims to address key issues in current systems, which are handling unstructured data, providing contextual understanding, and improving scalability. LLMs enable richer contextual comprehension, semantic analysis, and the user-centric personalization through their advanced natural language processing capabilities. In the study, we present a hybrid recommendation framework that combines machine learning, sematic similarity search, and Large Language Models (LLMs) through a Retrieval-Augmented Generation (RAG) pipeline. The dataset used in this study is obtained from Multimedia University (MMU). Key contributions include (i) identification of important factors affecting course recommendations, (ii) the development of a robust modeling framework using embeddings and LLMs, (iii) illustration of the advantages of LLMs including enhanced interpretability, contextualized recommendations and unstructured data handling. The findings suggest that LLM integration significantly improves precision, adaptability, and the transparency of course recommendations, indicating promising future developments in personalized education systems.

# Introduction

Course recommendation systems have become integral to the education sector, which help students to choose their academic pathways wisely. These systems assist students in finding programs that best fit their goals by evaluating information like academic background, individual preferences and demographics. Despite their usefulness, the existing course recommendation system faces several significant drawbacks. One major challenge is processing unstructured data. For example, free-form text like user reviews, personal statements, or open-ended queries which traditional systems often struggle to integrate with. Another limitation is that many recommendation models act as a “black box”, providing little insight into the rationale behind a suggestion and thus failing to offer clear justifications for their recommendations. Without transparency or explanations, users may find it difficult to trust or understand why certain courses are suggested to them.

To overcome these limitations, we explore the integration of Large Language Models (LLMs) into course recommendation systems. Unlike traditional models, LLMs such as OpenAI’s GPT-4 [1] or Meta’s LLaMA [2] can generate contextualized explanations in natural language, offering human-aligned justifications for each recommendation. These models possess language understanding, enabling them to interpret student information and course descriptions. By leveraging LLMs within the recommendation framework, we aim to make suggestions more interpretable and aligned with individual student background. In other words, the system can not only recommend courses but also explain why those courses suit the student’s profile, thus enhancing transparency and supporting better decision-making for students. Thus, this study proposes the integration of LLMs to course recommendation system to improve performance.

## Problem Statement

Despite advancements in educational technology, providing students with accurate and insightful course recommendations remains a major challenge. One of the primary difficulties lies in managing complex and diverse dataset that encompass various factors such as academic achievements, personality traits, demographics, and career aspirations. Many existing systems struggle to effectively integrate unstructured data which include open-ended responses, personal essays, and natural language queries with structured data. As a result, recommendations often end up incomplete, less personalized, or not sufficiently insightful. Moreover, traditional course recommendations models lack the adaptability required to generalize across a wide range of student profiles [3]. Typically, content-based or collaborative filtering techniques that rely on historical data and predefined relationships, fall short when encountering new student profiles. This limitation is evident in cold-start scenarios or when new courses are introduced into curriculum, where these models struggle to produce relevant suggestions [4]. Additionally, most current systems fail to consider the interdependence and contextual content among various academic and personal factors, limiting their ability to tailor recommendations to individual needs. Another significant issue is the handling of unstructured textual content such as personal statements, career goals and feedback [5]. Textual data often contains rich context about a student’s motivations and interests, yet many recommendation tools cannot extract meaningful insights from it. The lack of explainability in existing system further worsen eh problem as students are rarely given reason for why particular programs are suggested, leaving them disconnected from the decision process and reducing their trust and engagement [6]. Lastly, existing systems often struggle to adapt to evolving trends in user preferences, job market demands, or curriculum changes [7]. To provide relevant and future-proof guidance, a course recommendation system must continually update its knowledge base as educational landscapes evolves.

To address these challenges, the first objective of this study is to investigate different large language models as a recommendation system. Secondly, this paper aims to develop a hybrid course recommendation system using machine learning and similarity search. Thirdly, the last objective of this paper is to evaluate the performance of large language models.

# Related work

Large Language Models (LLMs) are advanced artificial intelligence systems that leverage deep learning and massive datasets to understand, process, and generate human-like text with high contextual awareness [8]. It represents a powerful fusion of Artificial Intelligence (AI) and Natural Language Processing (NLP), enabling them to develop a strong understanding of natural language. Transformer-based LLMs such as OpenAI’s GPT, Meta’s LLaMa and Google’s BERT brings an extraordinary era of advancement in machine learning and artificial intelligence [9]. Such models rely on billions of parameters to anticipate text, assess context, and perform a variety of language-based activities, such as conversational AI, code generation, translation, and summarization. For example, GPT-4 is one of the largest language models to date, boasting an astonishing trillions of parameters, demonstrating its vast complexity and capacity in language-related task [10].

In addition to generation capabilities, LLMs are also increasingly used for embedding generation. Embedding is a process that converts unstructured input into dense vector representations in high-dimensional space. These embeddings will capture the semantic meaning of input data and are crucial in powering vector similarity search and recommendation systems. In this study, LLMs are employed to generate sentences-level embeddings for user profiles and course descriptions using the “all-MiniLM-L6-v2” model from SentencesTransformer. After being generated, these embeddings are stored in a vector database to enable semantic course matching, allowing the system to retrieve courses most aligned with a student’s demographic and personality traits. Table 1 shows the summary of related works.

# Method

The proposed framework in Figure 1 begins with data preprocessing, feature engineering, and model training to generate course and student profiles’ embeddings. These embeddings are stored in a vector database for top-k course retrieval using similarity search.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TABLE 1.** Summary of related works | | | | | |
| **Author** | **Title** | **Algorithm** | **Dataset** | **Contribution** | **Evaluation**  **/Findings** |
| [1] | RAMO: Retrieval-Augmented Generation for Enhancing MOOCs Recommendations | * CF and CBF * RAG * LLMs * OpenAI Embeddings | Coursera Courses Dataset 2021 (Kaggle) | * Conversational explainability * Prompt engineering * Cold-start handling | Traditional CF and CBF algorithms failed when no prior user data was available |
| [10] | U-BERT: Pre-training User Representations for Improved Recommendation | * Transformer-based * Matrix Factorization * Deep Neural Networks | Amazon Product Reviews and Yelp | * Pre-trained user embeddings * Explainable Recommendations * Cold-start handling | MSE |
| [11] | From Interests to Insights: An LLM Approach to Course Recommendations Using Natural Language Queries | * RAG * Cosine Similarity Search * LLMs * OpenAI Embeddings | University Course Dataset (Course ID, Level, Title, Description, Precomputed Embeddings) | * Natural Language query interface * Cold-start handling | Successful sematic retrieval, fair recommendations, optimized retrieval rank |
| [12] | Supporting Student Decisions on Learning Recommendations: An LLM-Based Chatbot with Knowledge Graph Contextualization for Conversational and Mentoring | * LLM (GPT-4) * Knowledge Graph * Intent Classification * Cosine Similarity | Knowledge Graph (KG) with course metadata and relationships | * Conversational explainability * Knowledge graph * Prompt engineering | Classifier Accuracy, Precision, Recall and F1 Scores |
| [13] | Chat-REC: Towards Interactive and Explainable LLMs-Augmented Recommender System | * LLMs * k-nearest neighbor (kNN) * Matric Factorization * Prompt-Based | MovieLens 100k (943 users, 1682 movies, and 100k ratings) | * Conversational explainability * Cold-start handling | Precision, Recall, RMSE and MAE |

The dataset [14] consists of historical records of MMU graduations’ information. It consists of 2096 student records and 226 attributes per record. Key attributes include:

* **Demographic Variables**: Gender, race, nationality, and permanent address state.
* **Academic Variables**: Subject Grades (SPM), BM, and BI scores.
* **Course Attributes**: Program descriptions, faculty descriptions, campus branches, entry eligibility, job sectors, sponsor categories, and eligibility requirements.
* **Questionnaire Data**: Feedback on various university services such as library, labs, sports facilities, and cafeteria; ratings of lecturer interactions and innovative learning methods; perceptions of MMU’s reputation and counseling services; responses to open-ended questions about student preferences and concerns.

A diagram of a training process

AI-generated content may be incorrect.

**Figure 1.** Research Framework

## Data Cleaning and Preprocessing

The overall research framework is illustrated in Fig.1. The project begins with data cleaning and preprocessing, which prepares and standardizes the dataset by handling missing values, correcting inconsistent entries, and normalizing categorical fields. Following this, feature engineering is performed, where additional features related to students' personality traits (Openness, Conscientiousness, Extroversion, Agreeableness, and Neuroticism) are computed and added as new attributes. Before passing data into model training, the Synthetic Minority Oversampling Technique (SMOTE) is used to handle class imbalances to prevent models from being biased toward majority classes and improving prediction performance for underrepresented courses.

## Model Training

We first treat course recommendation as a multiclass classification problem to predict which program a student is most likely to enroll in based on their profile. The dataset is split 80/20 into training and testing sets. There are several machine learning classifiers trained on the structure data, which are Support Vector Machine (SVM), Random Forest (RF), K-Nearest Neighbors (KNN), Decision Tree, Logistic Regression, and XGBoost. During training, each classifier is evaluated using Accuracy, Precision, Recall, F1-socre, and Confusion Matrix to select the most performing model.

## Embeddings and Top-k Retrieval

To enable semantic matching, we use the “all-MiniLM-L6-v2” model from SentenceTransformer to generate embeddings for student profiles and course descriptions. These embeddings (384-dimension) are stored in a vector database, enabling semantic similarity searches. Upon receiving a new student profile, the system will generate a sentence embedding for student. A Top-20 semantic retrieval is performed using vector database to identify the 20 most relevant course embeddings. These 20 course descriptions, along with the student’s profile traits are passed into a structured prompt for LLM.

## Prompt

The prompt given to LLM is: “*Here are 20 available study program, recommend the most suitable 5 courses for this user. Do NOT create or suggest any new courses outside this list. For each selected course, explain in 2–3 sentences why it matches the user's traits. Follow the numbering (1 to 5) based on your ranking*.” This prompt ensures LLM focuses strictly on the provided courses and. This prompt is processed by a Large Language Model (LLaMA-3), which serves as the reasoning engine. The LLM evaluates the 20 candidates considering the user’s profile, selects the five most suitable courses, and generates a human-readable justification for each selected recommendation. Lastly, the system outputs this ranked list of five university courses tailored to the student’s profile, leveraging both the preliminary machine learning predictions and the semantic alignment achieved through LLM-driven reasoning.

## LLM Integration and Recommendation

The LLMs considered in this project include GPT-based models, and Ollama. The GPT-based model consists of GPT-3.5 Turbo and GPT-4, while Ollama model includes Llama2 and Llama3. Based on their performance, speed and cost effectiveness, GPT-based models offer better performance with faster response times, making them highly efficient for real-time applications [1]. However, they require paid API access, which increases long-term operational expenses. In contrast, Llama models are completely free and open-source, but they have slower response times compared to GPT-based models [9]. Despite the slower inference speeds of Llama models, even though GPT models are faster, the cost of paid models is a limiting factor, and for a budget-friendly recommender system, Llama models are the best choice. After the LLM processes the prompt, it produces a ranked list of 5 courses along with a justification for each selection. The final recommendation output presented to the user is the refined list. The system leverages RF for initial predictive insight, the semantic vector search for content-based matching, and LLM for reranking and reasoning. By combining these components, this framework predicts suitable courses and explains them in human-readable language. The overall system is evaluated in two modes: one without LLM (using only the semantic similarity ranking), and one with the LLM integrated to observe the differences in the recommendations.

# findings

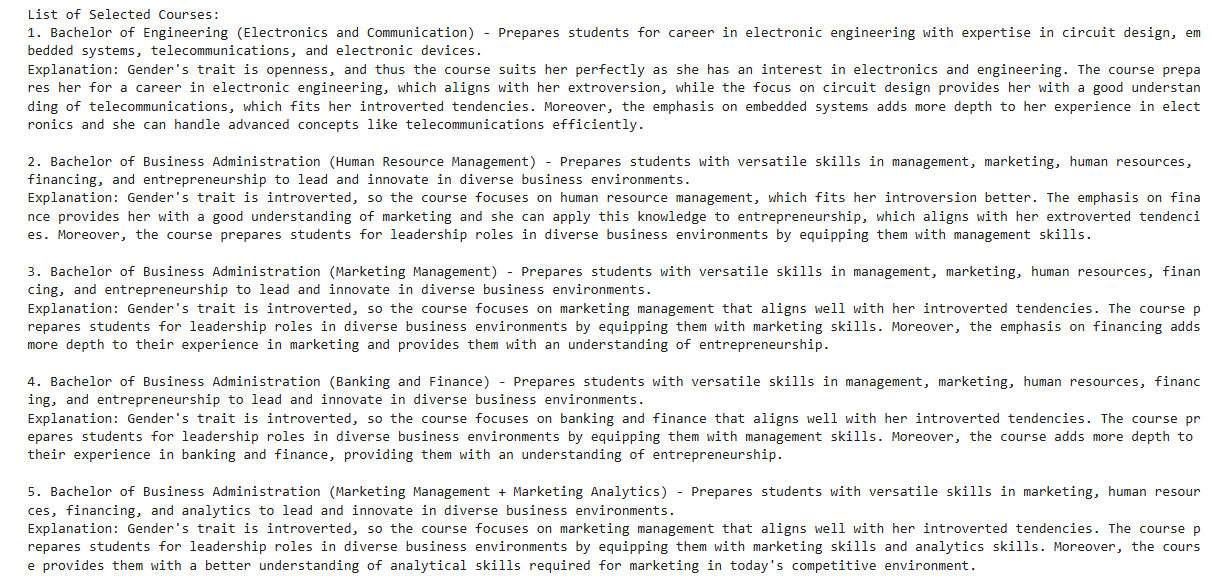
In this study, several machine learning classifiers are evaluated based on their ability to classify data accurately. After training, each model is evaluated on the historical student data, and performance is assessed using four key metrics, which are Accuracy, Precision, Recall, and F1-score. The results are summarized in Table 2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TABLE 2.** Summary of the performance of classifiers | | | | |
| **Classifier** | **Accuracy** | **Precision** | **Recall** | **F1-score** |
| SVM | 0.57 | 0.55 | 0.58 | 0.55 |
| **Random Forest** | **0.80** | **0.79** | **0.80** | **0.79** |
| K-nearest Neighbors | 0.72 | 0.70 | 0.72 | 0.70 |
| Decision Tree | 0.66 | 0.65 | 0.66 | 0.65 |
| Logistic Regression | 0.31 | 0.26 | 0.31 | 0.27 |
| XGBoost | 0.78 | 0.77 | 0.78 | 0.78 |

Based on the performance comparison in Table 2, Random Forest (RF) classifier achieves excellent performance with the highest accuracy of 0.80, precision of 0.79, recall of 0.80 and F1-score of 0.79. XGBoost is a close second with an accuracy of 0.78, but it slightly underperforms RF on all metrics. K-nearest Neighbors classifier achieves accuracy of 0.72, it performs well but does not surpass ensemble methods. Other models like Support Vector Machine (SVM), Decision Tree show average performance but are not as strong as RF. Notably, the Logistic Regression model performs quite poorly, with only 31% accuracy and an F1-score of 0.27, indicating that a simple linear model is insufficient for this complex multiclass task. Based on the result, RF is selected as the best model for further implementation, as it outperforms all other models in every metric. With the RF chosen as the baseline predictor, we next examined the effect of integrating semantic search and LLM reasoning into the recommendation process. We constructed an example to illustrate the LLM-augmented recommendation and to compare it with the non-LLM approach. After integrating LLM, the system analyses a student’s profile alongside the retrieved course descriptions and provides a detailed justification for each suggestion.

Figure 2 shows example output for a sample student profile (Female, Selangor, Chinese ethnicity, high Openness and Conscientiousness) with the prompt ("*Here are 20 available study program, recommend the most suitable 5 courses for this user. Do NOT create or suggest any new courses outside this list. For each selected course, explain in 2–3 sentences why it matches the user's traits. Follow the numbering (1 to 5) based on your ranking*.”). In this scenario, the system first retrieves top 20 candidate courses for the student via semantic similarity search. The LLM is then prompted to rank those courses and explain why the 5 selected course matches the user’s traits. The LLM-generated output consists of ranked list of courses, each accompanied by a brief explanation. These explanations explicitly reference the student’s characteristics and connect them to aspects of the course. The result is more personalized recommendation list where the student can see not just what is recommended, but why those courses are recommended.

To highlight the impact of LLM integration, Table 3 provides a side-by-side comparison between the semantic search with and without LLM component. Table 3 highlights the qualitative improvements after integrating the LLM. In the first stage without LLM, the system would retrieve the top five course options for the student based solely on vector similarity to the student’s profile. This approach did not provide any deeper analysis or explanation for the user. The recommendations were essentially a list of course titles ranked by relevance score. In the second stage with LLM, the system produces personalized and explainable recommendations.



**Figure 2.** Course recommendation output after LLM integration

|  |  |  |
| --- | --- | --- |
| **TABLE 3.** Comparison between semantic search with and without LLM | | |
| **Aspect** | **Semantic Search without LLM** | **Semantic Search with LLM** |
| Recommendation Style | Top 5 ranked courses purely based on similarity score | Top 5 courses are selected based on both semantic matching and interpreted via user personality traits. |
| Interpretability | No explanation provided; only course names and descriptions listed. | Detailed paragraph explanations aligning course suggestions to the user’s openness, conscientiousness, and career aspirations. |
| Transparency | Limited - user only sees course title and score. | High - users receive clear justifications enhancing trust in recommendations |

# CONCLUSION

This project presents a hybrid course recommendation framework combining ML, semantic vector retrieval, and LLM-based explanation. It integrates RF classification with 80% accuracy on historical records, the embedding-based vector search for relevant courses even for new profiles, and the LLM for natural language justifications. The results demonstrate that LLM-augmented system outperforms a purely similarity-based recommender in terms of contextual matching and transparency. These findings confirm that incorporating LLMs can greatly improve the precision, personalization, and interpretability of course recommendations compared to traditional methods.

More broadly, this study suggests that educational recommender systems can become more student-centric by combining predictive analytics with the contextual capabilities of modern LLMs. The approach of using LLMs to connect user traits to course has potential applications in other domains where explainable recommendations are critical. At the same time, we find out key considerations for real-world deployment. Addressing issues of fairness, mitigating potential biases and managing the computational cost of LLMs are essential. Future work includes scaling evaluation to larger student datasets, systematically assessing fairness and robustness, and optimizing LLM performance via prompt refinement or efficient open-source models. This study highlights a direction for the future of course recommendation system where a system that not only delivers accurate recommendations, but also clearly communicates the reasoning behind them. This empowers students to make better decisions about their academic paths.

# References

1. OpenAI, J. Achiam, S. Adler, S. Agarwal, et al., “GPT-4 Technical Report,” (2023).
2. Meta AI. (2024). The Llama 3 Herd of Models. arXiv preprint arXiv:2407.21783.
3. J. Rao, and J. Lin, “RAMO: Retrieval-Augmented Generation for Enhancing MOOCs Recommendations,” arXiv preprint arXiv:2407.04925 (2024).
4. J. Ji, Z. Li, S. Xu, W. Hua, Y. Ge, J. Tan, and Y. Zhang, “GenRec: Large Language Model for Generative Recommendation,” in Advances in Information Retrieval, edited by N. Goharian, N. Tonellotto, Y. He, A. Lipani, G. McDonald, C. Macdonald, and I. Ounis, (Springer Nature Switzerland, Cham, 2024), pp. 494–502.
5. L.M. De Campos, J.M. Fernández-Luna, and J.F. Huete, “An explainable content-based approach for recommender systems: a case study in journal recommendation for paper submission,” User Model User-Adap Inter **34**(4), 1431–1465 (2024).
6. N. Afreen, G. Balloccu, L. Boratto, G. Fenu, F.M. Malloci, M. Marras, and A.G. Martis, “Learner-centered Ontology for Explainable Educational Recommendation,” in *Adjunct Proceedings of the 32nd ACM Conference on User Modeling, Adaptation and Personalization*, (ACM, Cagliari Italy, 2024), pp. 567–575.
7. S. Amin, M.I. Uddin, A.A. Alarood, W.K. Mashwani, A.O. Alzahrani, and H.A. Alzahrani, “An adaptable and personalized framework for top-N course recommendations in online learning,” Sci Rep 14(1), 10382 (2024)
8. L. Xu, J. Zhang, B. Li, J. Wang, S. Chen, W.X. Zhao, and J.-R. Wen, “Tapping the Potential of Large Language Models as Recommender Systems: A Comprehensive Framework and Empirical Analysis,” arXiv preprint arXiv:2401.04997 (2024).
9. Y. Chen, Y. Yan, Q. Yang, Y. Shu, S. He, and J. Chen, “Confidant: Customizing Transformer-based LLMs via Collaborative Edge Training,” arXiv preprint arXiv:2311.13381 (2023).
10. Z. Qiu, X. Wu, J. Gao, and W. Fan, “U-BERT: Pre-training User Representations for Improved Recommendation,” AAAI **35**(5), 4320–4327 (2021).
11. H. Van Deventer, M. Mills, and A. Evrard, “From Interests to Insights: An LLM Approach to Course Recommendations Using Natural Language Queries,” arXiv preprint arXiv:2412.19312 (2024).
12. H. Abu-Rasheed, M.H. Abdulsalam, C. Weber, and M. Fathi, “Supporting Student Decisions on Learning Recommendations: An LLM-Based Chatbot with Knowledge Graph Contextualization for Conversational Explainability and Mentoring,” arXiv preprint arXiv:2401.08517 (2024).
13. Y. Gao, T. Sheng, Y. Xiang, Y. Xiong, H. Wang, and J. Zhang, “Chat-REC: Towards Interactive and Explainable LLMs-Augmented Recommender System,” arXiv preprint arXiv:2303.14524 (2023).
14. Ministry of Higher Education Malaysia, "Sistem Kajian Pengesanan Graduan (SKPG)," 2022. [Online]. Available: https://graduan.mohe.gov.my/skpg24/.