International Journal on Advanced Science Engineering Information Technology

Vol.13 (2023) No. 5
ISSN: 2088-5334

College Course Recommender System based on Sentiment Analysis

Naufal Zamri [a], Naveen Palanichamy [a,*], Su-Cheng Haw [a]

[a] *Faculty of Computing and Informatics, Multimedia University, 63100, Cyberjaya, Malaysia*
Corresponding author: [*]p.naveen@mmu.edu.my

Abstract— **College plays a vital role in defining a student's future by providing relevant education, skills, and exposure. The choice of college courses heavily influences their career foundation and employment skill sets. However, the expanding number of college courses often leaves students struggling to make the best choice, leading to dropouts due to the lack of interest. Many systems rely on existing student reviews or the popularity of the course itself, which may not always result in relevant recommendations. Hence, some systems use sentiment analysis (SA) to evaluate students' opinions, considering qualitative and sentiment data to understand their interests better. However, current SA performance struggles to extract meaningful words due to dataset availability. Hence, a course recommendation system based on students' interests and competence would be valuable. This paper focuses on evaluating and understanding existing systems to provide students with an effective course recommendation system. It includes first gathering useful data that would improve the use of SA. Next, feature extraction techniques Term Frequency-Inverse Document Frequency (TF-IDF) and N-gram were implemented and compared. SA will be performed to increase the relevance of the student's interests to recommend a course by implementing Fuzzy Logic and K-nearest neighbors. These algorithms will be evaluated by performance metrics such as accuracy to determine the most efficient way to recommend a course. The findings highlight the importance of considering students' subjective preferences and interests for better outcomes regarding student success and graduation rates.**

Keywords— **Recommendation system; natural language processing; sentiment analysis; K-nearest neighbor; fuzzy logic.**

Manuscript received 18 Dec. 2022; revised 14 Jun. 2023; accepted 20 Aug. 2023. Date of publication 31 Oct. 2023.

I. INTRODUCTION

Choosing a college course is an important decision to begin their career path. After completing their Foundation or Diploma, students are required to make an important decision: choose a suitable Degree course of study in college. While some are determined to select a specific course, many are struggling to decide what the best course is for them. In addition to the abundance of choices offered in college, it is plausible for students to receive an overwhelming number of choices without proper guidance [1]. Students may have limited experience or knowledge about a particular course they might find most suitable if only they knew. Hence, a course recommender who could guide and navigate them to select a suitable course would be helpful.

Using quantitative data, most existing recommendation systems can suggest a course to the user. This includes reviewing existing student ratings of a given course. This information restricts proper evaluation of the student's interests as the option is limited [2]. In addition, recommending a course to another student who has already taken the same course may be irrelevant as the recommendation is tailored toward another person. Yet, the greatest challenge faced regarding tailored recommendation systems is cold start data [3]. Thus, the recommendation must be based on personalized and custom attributes that define the user's interests.

This paper will discuss several important aspects of the college recommendation system. The paper begins with discussion on relevant studies related to course recommendations. There are three types of recommendation systems: content-based filtering (CBF), collaborative filtering (CF), and hybrid filtering; a combination of content-based and collaborative filtering. In the existing recommendation systems, CBF was found helpful when determining a suitable course if it is similar to what the user has taken before. This includes comparing similar professors, competence requirements, student's knowledge area, as well as the subject contents of the course itself [4]. In contrast, CBF was known for its qualitative approach to neighbor users with similar characteristics in order to obtain further information about them [5]. However, CF struggles with analyzing new users with no information or course evaluations to a neighbor with other users as opposed to CBF, which handles well with cold start problems. However, a pre-defined dataset is required to match the user's information regarding CBF. This would be an issue should the user provide new information that is undefined to the system's dictionary [6]. In addition, CBF's descriptive approach to recommending a user based on its similar characteristics with the course limits qualitative data. Thus, an additional sentiment approach would be beneficial to compensate for the gap.

Many researchers would use qualitative methods to further analyze the user's needs to obtain a high-relevance recommendation. Thus, extracting important information from the user's reviews and course information. Some of the most prevalent and common techniques include C-BOW, Skip-Gram, Latent Dirichlet Allocation (LDA), Matrix Factorization (MF), Ontology, N-gram, and Inverse Document Frequency.

Continuous Bag of Word (C-BOW) and Skip-Gram (S-G) are effective word embedding techniques. C-BOW is mainly used to predict ratings on given words, while S-G predicts descriptions using topics, particularly for rare words. Ezaldeen's findings suggest that C-BOW outperforms S-G in predicting common and frequent words in E-Learning, achieving an accuracy ratio of 76.2%: 84.9% [7]. Moreover, the N-gram model algorithm is utilized to enhance other text analysis techniques. A course recommendation model based on the N-gram classification technique was proposed to cater to scholars' needs and interests [8]. The N unit determines the number of words analyzed at a time, with a higher N unit resulting in more overlapping data [9]. Setting the N unit to 3, Trigram is proposed in this research, employing a sequence of measurements such as (z1, z2, z3), (z2, z3, z4), . . ., (zp−2, zp−1, zp-). Ontology can also be employed with N-gram by defining general topics extracted from the words [3].

Additionally, the MF technique decomposes features into matrices to identify patterns and topics in the text, allowing course recommenders to overcome data sparsity issues and make recommendations based on item-rating patterns [1]. In other words, it represents the frequency of selected words in columns and their occurrences in comments in rows [10]. TF-IDF is employed to extract features from text by reducing redundant information, considering a word's importance based on its occurrence frequency [9]. The LDA algorithm is then utilized for topic modeling, analyzing the extracted words to identify the distribution of different topics represented in course descriptions. These keywords are compared with the student's feedback obtained through LDA, helping to determine the course being described [1]. LDA is valuable for uncovering underlying topics and can be implemented using existing libraries like Gensim and IdaMallet [2]. A technique utilizing the LDA user interest model was developed for recommending online education courses by evaluating user preferences and interests in topics [11].

The recommender system [12] clusters jobs using K-Means and TF-IDF algorithms and then determines similarity and dissimilarity using the Cosine Similarity algorithm. The study [13] showcased the recommendation system modeling experiment findings for personalized learning using the TF-IDF and the TF-IDF algorithm and Cosine Similarity. Table 1 summarises the different feature extraction and topic modeling algorithms used by different researchers.

TABLE I
SUMMARY TABLE OF FEATURE EXTRACTION

Author	C-BOW	S-G	LDA	MF	Ontology	N-Gram	TF-IDF
[1]			√	√			
[2]			√				
[3]					√	√	
[4]					√	√	
[6]			√				
[7]	√	√					
[8]						√	
[9]						√	√
[10]				√			√
[11]			√				
[12]							√
[13]							√

Out of all the discussed techniques used, TF-IDF, N-gram and LDA are one the most used by researchers. However, there has not been any research to evaluate and compare these 3 algorithms together. Given that N-gram and TF-IDF have been experimented on a prediction model and were proven to provide a high result, it might be beneficial to apply the hybrid algorithms to a different prediction model [9].

Natural Language Processing (NLP) is widely used in recommendation systems to extract relevant information from user input. Many researchers have explored Sentiment Analysis (SA), a text-mining technique that helps define the user's emotions towards a topic. SA helps evaluate qualitative data to enhance the relevance and accuracy of the prediction. In course recommendations, SA is implemented to review evaluations from students and professors regarding the courses they are taking. Sentiment words are extracted and are used to rate the quality of the course. Ng suggests that the length of the text will affect the score as it is expected to have more sentimental words in it [1]. SA can be implemented using built-in libraries available in Python, such as TextBlob, Flair, and VADER. Algorithms like TextBlob can be used as text labeling to categorize data into "Positive", "Negative", and "Neutral" class based on the polarity of the text [14]. Natural language processing techniques and semantic analysis were utilized for course recommendation in e-learning, while machine learning analysis improved user ratings in the e-learning environment by automatically analyzing learner characteristics and learning styles through clustering strategies [15]. However, with defined libraries, some sentiment models misclassify the reviews and label them into a category they do not belong to. It is suggested to have an additional algorithm, such as Lexicon, to consider different sentiments more thoroughly [2].

The final procedure of the recommendation system is to predict a suitable course for the user. This is done by matching the user input with the course itself. Different authors suggested several prediction models. Machine Learning certainly becomes essential for the system to make precise decisions. This includes Convolutional Neural Network (CNN), Fuzzy Logic, K-Nearest Neighbour, and Decision Trees. Firstly, CNN is used for classification, categorizing texts into meaningful categories and predicting based on sequence similarities. It incorporates sentiment analysis to assess course feedback [16]. CNN requires a large training set to improve its classification accuracy by recognizing patterns over time. In E-learning recommendation systems, CNN can predict ratings using textual sentiment values [17]. The research recommends a layered CNN implementation with word embedding, resulting in a CNN-two-channel model achieving 77% accuracy. Vanessa suggests combining CNN with LSTM to eliminate manual feature extraction. This is because CNN finds important patterns that simplify the process and improve accuracy without requiring humans to pick out the necessary features [18] manually.

Fuzzy Logic provides an approximate representation of an output, making it suitable for subjective tasks like course recommendations. It effectively handles uncertainty and imprecision in data, enabling the model to make informed decisions. Fuzzy Logic is instrumental when dealing with complex and irregular student information. A recommendation system based on fuzzy Logic and machine learning is described in [19] for determining the most beneficial academic program for students. The system uses a student dataset with 21 features and 1000 individual cases and employs attribute selection methods and machine learning techniques such as fuzzy SVM, random forest, and C4.5 to predict the most suitable academic program for students. A course recommendation system [20] addresses the cold start issue of the recommender system by using the Mamdani Fuzzy Inference System to classify skill levels and suggest equivalent courses based on student interests and abilities.

In addition to that, the K-Nearest and Neighbors (KNN) algorithm allows better visualization of which class is most relevant to an item. KNN determines which course is most similar to the user based on the user profile. This can be done by first measuring the distant metrics in which the research applies cosine similarity [21]. The k value inside the model plays a crucial role. A small k can make predictions based on noise and outliers, potentially leading to overfitting, while a large k value is useful to reduce noise or outliers [22]. In addition to that, it also measures which course is most similar or related to each other. This way, it can recommend relevant courses available. Figure 1 illustrates an example of course neighbours produced using KNN.

KNN is very suitable for classification and prediction, especially when comparing two or more data from each other. When it comes to course recommendation, KNN can be used in many ways, from finding the nearest course based on the user's preference and finding similar courses that are related to each other [23]. In [24], the KNN and Feature weighted algorithms were utilized to present the top N comparable clients for the test clients and suggest the Top M colleges to clients from the N comparative clients. Based on historical data and user preferences, [25] proposed a recommender system that utilizes collaborative filtering methods like K-nearest neighbour (KNN), Singular Value Decomposition (SVD), and neural network-based collaborative filtering (NCF) models to recommend e-learning courses by analyzing a dataset of one lakh Coursera course reviews from Kaggle.

Decision Tree, along with other tree-based models are employed as a method to classify users based on their course preferences. A system interacts with users, guiding them to build their learner's profile by segmenting their information based on their responses. This classification helps recommend suitable learning courses, addressing the issue of information overload and providing guidance during the decision-making process [26], [27]. Additionally, the classification model is supported by prediction models like KNN.

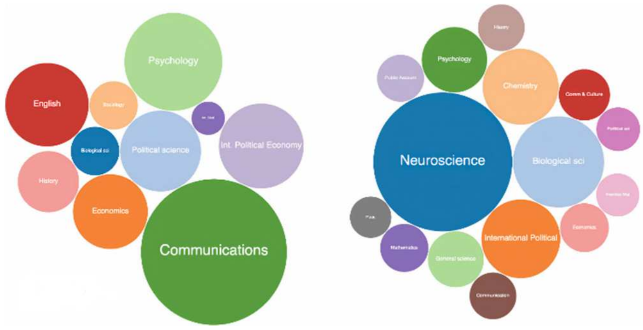

Fig. 1 Nearest Neighbour Related Course Results [21]

K-means is an unsupervised machine learning algorithm used for clustering similar data or words based on their features. It is valuable in text classification and topic modeling for NLP, identifying topics in documents. K-means can also group learners with similar attributes using the genetic K-means algorithm (GA K-means) [16]. GA K-means compensates for the disadvantages of the traditional K-means algorithm, such as computational expense and the dependence on initial center selection. It overcomes computational complexity through real-time implementation. Table 2 summarises the different techniques researchers use to predict a suitable course.

TABLE II
SUMMARY TABLE OF PREDICTION MODEL

Authors	CNN	Fuzzy Logic	K-NN	Decision Trees	K-Means
[3]		√			
[4]					
[5]		√			
[7]	√	√			
[10]		√			
[16]					√
[17]	√				
[18]	√				
[19]		√			
[20]		√			
[21]			√		
[22]			√	√	
[23]			√		
[24]			√		
[25]			√		
[26]			√	√	
[27]			√		
[28]		√			
[29]		√			
[30]		√			
[31]		√			

Several algorithms are worth taking into consideration. Firstly, Fuzzy Logic works well in an irregular prediction pattern as the results are based on approximate representation, which is appropriate for a recommendation system [28][29]. Fuzzy logic systems are said to be easy to understand, but there is a great challenge when there are various input variables, as there are a number of rules to be set, making the system harder to explain and use [30]. Sharma suggested modifying the traditional fuzzy Logic into meditative fuzzy Logic to handle contradicting and evolving information to ensure its consistent predictability [31]. Both Fuzzy Logic and KNN algorithms are great tools for model selection. The research papers found that comparing these two models has not yet been discussed to determine which model provides a better result depending on the same requirements and tasks.

In summary, this section unravels different techniques that can be used to provide an appropriate recommendation to the user. The most common and effective approach would have been to evaluate more qualitative rather than quantitative data from the user to enhance the recommendation prediction. To achieve this, sentiment analysis has become the main component inside their system to help evaluate the person's interests. With many procedures conducted by research, the most important step is data preparation. Many proposed systems struggle with data sparsity, or the collected data contains mainly quantitative data. In addition, the data collection to recommend a suitable course is based on existing data found online, which is not personal and relevant to the current user's specific needs. A more interactive and customized approach to data collection may be essential before extracting and evaluating the texts to build upon the research gaps of previous models.

While most systems mainly focus on a single algorithm to extract data and predict outcomes, this research inspires the need to investigate the algorithms performed and determine whether they can be used in a novel college course prediction system. In addition, given the lack of comparison between high-accuracy algorithms and NLP techniques, this research also aims to extend the experiment of using different approaches to Determine the most suitable method to recommend students with an appropriate college course.

II. MATERIAL AND METHOD

This section describes the procedures and implementations of the course recommendation system. This includes the data sets, the data preparation steps, the algorithms used to determine the college majors to recommend for a student, and the metrics used to evaluate the results. Figure 2 illustrates the system flow and the techniques for producing a course recommendation.

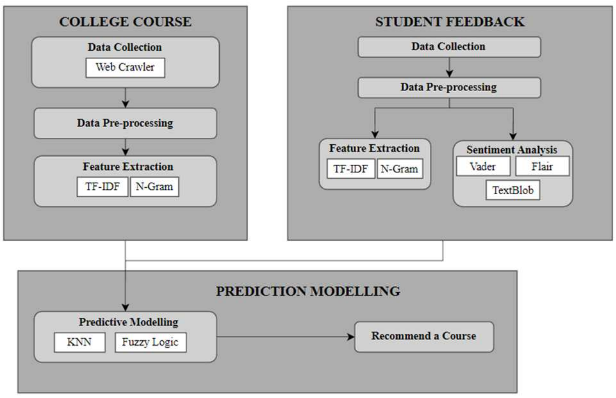

Fig. 2 System Flow of Course Recommendation System

In general, this paper aims to measure the best performance for a college recommender by implementing a content-based recommendation system supported by Sentiment Analysis. The system will begin by collecting data from the student's feedback and applies feature extraction techniques such as TF-IDF and N-gram to determine the topics relevant to the user's interests. With the support of sentiment analysis, the system can narrow down the topics based on the user's most preferred choice. This research also aims to compare two different prediction models, K-Nearest Neighbour and Fuzzy Logic, to evaluate which models should be used to recommend a suitable college course to the user.

A. Data Collection

This system uses two datasets to predict a recommendable college course. The first dataset used is a list of college courses and their details. Given that this paper aims to recommend Foundation and Diploma students with a degree course, the dataset contains a combination of degree courses available from different universities in Malaysia. The data contains the course name, description, the subjects available, career prospects, and its minimum requirements. College course information can be retrieved from the university websites themselves. In this case, courses available at Multimedia University, Taylor's University, Asia Pacific University, Uniten, and Universiti Sains Malaysia are explored. The websites contain brochures and online information containing the relevant information. Using Web Crawling in Python, the data scraping process has become far more efficient. However, given that the format of the websites may vary, some contents need to be verified and scraped manually to ensure proper data was taken. Although an abundant list of courses is available with similar attributes, this dataset aims only to include courses available in at least two of the chosen universities. In total, there are 34 college majors offered in 2023 collected across different faculties and fields of study.

The second data set contains the student feedback records, where each record describes the student's interests in taking a course they have in mind when they enter a degree program. This data was obtained as a survey, targeting Foundation and Diploma students in Malaysia to fill in the form. To protect the privacy and confidentiality of the student feedback data, sensitive information such as their personal details was not asked, as the only data required is their opinions regarding college courses and their performances during their Foundation or Diploma program. The survey also acts as a guidance system prototype for the students to choose the suitable course. This is because the questions asked to the students are customized depending on their choice of answers from the previous question. This helps narrow the abundant course options by starting with a general overview of the student's field of interest and slowly narrowing down to a more specific question. The user is allowed to evaluate two fields of interest, which will be analyzed later. The delivery of the student feedback is using Google Forms as it is easily accessible for young students to fill out the form on any device.

As seen in Figure 3, Google Forms allows the feedback to have a custom set of questions depending on the user's response. In most of the questions asked the user is asked to write a lengthy sentence to express their opinion related to a certain topic. This allows more qualitative responses, which will enhance the level of sentiments provided by the user.

With the student feedback data collected, this paper will be using 70% of the responses as the training set and 30% as the testing set.

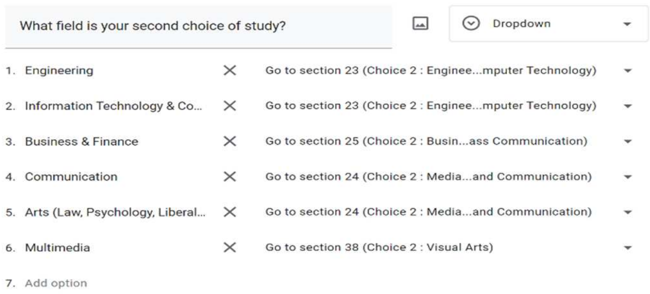

Fig. 3 Student Feedback Google Form

As the raw dataset consists of multiple attributes depending on the options the user has chosen, many null values exist as the user did not choose that section. Thus, the dataset must first be cleaned by merging the user's responses into three attributes: their 'General Opinion', their evaluation of their 'First Option' of the study field, and their 'Second Option'. Each text category is compiled from the users regarding a field of study. The dataset is then manually labeled based on the user's responses in the two fields of study. This would be used as a reference when the system predicts a suitable course for the user.

B. Data Pre-processing

Once data has been collected, an important step is to clean the information before extracting important words from both the course description and the student feedback. Several things are worth cleaning before the system undergoes feature extraction. There are, however, different pre-processing features implemented in the college course dataset compared to the student feedback. This is because the college course dataset aims only to contain relevant and useful information regarding the course, whereas it might be possible that removing some words in the student feedback may lose its sentiment value. The texts are first converted into lowercase in the college course dataset to standardize all the words. This is then followed by removing punctuation and ready to be tokenized.

1) Tokenization: Tokenization is simply the process of breaking down the text into a list of words. This broken down of words is considered tokens, whereby pre-processing and future feature extraction evaluate each token's value instead of the entire text document. This will be helpful later when measuring the weight of each token during the feature extraction process.

2) Stop Words Removal: Stop words are common prepositions and phrases used to complete a sentence. Words like this are considered irrelevant for extraction as they do not bring any valuable information to categorize in the course prediction. Thus, removing these words before analyzing the text is best. Stop words are logged inside a built-in dictionary found in Python. In total, there are 179 stop words stored. This includes "the", 'where', 'do', "you've", 'and' etc. Should any of these stop words be found in a text, it is best to remove them first to only consist of relevant and useful words. Despite the benefits of the words provided, some common stop words are not included in the dictionary. Thus, additional words are also included inside the stop words dictionaries to enhance the filtration. Stop words were removed from the college course dataset to further help extract only relevant words that best describe the course. Some common college terminologies are found throughout different course descriptions used to describe the course structure. Given that some of the terminologies describe the program's structure and not the course content, the words are added inside the stop-word removal as they do not bring significant value to the course description.

3) Lemmatization: When analyzing and classifying words in a text, there are many inconsistencies in the grammar that turn out to have the same meaning. This occurs when a word is branched and modified into different types of words, such as adverbs, plural, and nouns. Lemmatization is the process of reducing different forms of words into a single form. Standardizing this would ease analyzing the text in the future. For example, words like "creates", "creating", "created" and "creation" will be reduced to "create". To achieve this, Part-of-Speech tagging (POS tag) helps assign a label to each word read to indicate its grammatical category. Lemmatization is applied to standardize the pre-processed text to ensure all stop words are correctly verified. Once lemmatization is implemented, stop words are eliminated again so the converted words are properly processed.

C. Feature Extraction

Once the texts have been pre-processed, the system could undergo the feature extraction process. This is to extract relevant keywords from both datasets. Inside the college course dataset, feature extraction is applied to set a list of keywords to describe the college course. On the other hand, the student's feedback aims to extract keywords that could describe the courses they are referring to in order to match with keywords found from the college courses.

TABLE III
FEATURE EXTRACTION EXPERIMENTS USING DIFFERENT ALGORITHMS

Method	Words Extracted from Civil Engineering Description
TF-IDF	activity, environmental, structure, nature, integration, geotechnical, civil, undertake, diverse, design, demonstrate, conceptual, competency, practical, ethical, analysis, civil, structural, steel, society, practical, mechanic, hydraulic, geotechnics, geology, engineer, concrete, technology, management, equation, academician, construction, transportation, system, structural, operation, material, wastewater, instrumentation, highway, engineer, development, consult, research, design
TF-IDF and N-gram	integration, problem, instill, integrate, activity, life, management, multidisciplinary, namely, nature, nurture, time, preliminary, principle, produce, advanced, practical, hydrology, integrate, laboratory, management, material, mechanic, method, numerical, personal, reinforce, highway, society, statics civil, wastewater, highway, system, operation, material, manufacture, transportation, instrumentation, structural, station, engineer, design, control, consult, construction

The system applies TF-IDF as a feature extraction technique. In this system, the maximum number of features extracted, n, is set to 15. The system first transforms the text to generate the TF-IDF matrix. Once this is accomplished, to generate the extracted features, Python's panda could display the top 15 features. This subsection focuses on experimenting with two feature extraction algorithms and examining the quality of results should N-Gram be added inside the TF-IDF current function. In this paper, the 'N' value is set to 3, or is known as Trigram. A sample result of the feature extraction can be seen in Table 3.

By default, TF-IDF only considers individual words when there are times when the meaning of a word is clearer when paired with other words. Hence, N-grams come in handy to combine with TF-IDF to improve the accuracy of the TF-IDF word extraction process. However, as seen in Table 3, the words extracted from TF-IDF using the trigram approach are similar as when using TF-IDF only. This could be due to the specific characteristics of the dataset and the text being processed. It could potentially be because the dataset of the college course description does not contain many distinct trigrams or if the trigrams do not significantly contribute to the overall TF-IDF scores, which is why the extracted words are like the ones obtained using TF-IDF without trigrams.

It may be worth exploring different values when adjusting the TF-IDF and N-gram values, such as increasing or decreasing the number of maximum values extracted. The value of the N-gram can also be modified to either a bigram or an increase in the number of N.

D. Sentiment Analysis

In this subsection, the application of Sentiment Analysis in the system will be explained thoroughly. Sentiment Analysis is applied to measure the sentiment score of each topic the user discusses. With several topics the user mentions about their preferences, using Sentiment Analysis would help the system determine which topic sparks their interests the most. Thus, the system can recommend a suitable college course based on the topic in which the user has the highest sentiment value. Sentiment Analysis achieves this by measuring the number of positive sentiment words found in comparison to the total sentiments.

Several built-in libraries from Python can be implemented to measure the sentiment score of the user's feedback. Using a lexicon-based approach, VADER can be used to measure the intensity of the sentiments found inside the text. The value is normalized to a range between -1.0, considered the most negative sentiment score, and 1.0, the most positive sentiment score. As seen in Table 4, VADER measures the polarity of the entire user response and summarizes the sentiment score. If most of the words in the text are mainly positive, then the polarity value would be high. In this case, it is not necessary to categorize the polarity score as 'Negative', 'Positive', 'Neutral'. This is because given that the questions asked to the user mainly focus on the courses they enjoy pursuing, the sentiment score would most likely be a positive value. From here, the system can determine between the two choices of unknown course the user is describing and which is better to be recommended.

It is also worth exploring different built-in libraries that can perform sentiment analysis. TextBlob trains the student feedback based on Naïve Bayes. By tokenizing the text, it can calculate the polarity score ranging from -1.0 to 1.0 like VADER. On the other hand, Flair applies deep learning models to perform sentiment analysis and provides more advanced capabilities by embedding the text and classifying sentiment at a sentence level. As seen in Figure. 4, different libraries provide different sentiment polarities, which would affect the system's decision in choosing the topic the user is most interested in.

TABLE IV
VADER SENTIMENT POLARITY SCREENSHOT SAMPLE

ID	First Choice	First Choice vcompound	Second Choice	Second Choice vcompound
17	I like to learn about designing and creating …	0.7469	Information Technology & Computer Science …	0.9643
18	I like to apply my mathematical and problem...	0.8271	Communication. I like to learn about organisa …	0.8720
19	I like to learn about programming and how …	0.8313	Multimedia. I want to be an animator.animation …	0.8658
20	I like to strategize my communication through…	0.8807	Art (Law, Psychology, Liberal Arts). I like to …	0.7184
21	I like to apply mu mathematical and problem …	0.8720	Business & Finance. Choice 3. I am very good at …	0.9722

Fig. 4 Sentiment Analysis Method Comparison

VADER is well-suited for analyzing informal language, making it a good fit for student feedback. Compared to Flair or TextBlob, VADER tends to be faster in its processing and more efficient as it is based on pre-defined lexical features. In addition, as seen in the graph above, the sentiment score has a much more consistent value, which allows the choice of words to be as sensitive as Flair or TextBlob. Thus, making it more reliable to compare and analyse the sentiment scores.

E. Predictive Modelling

This subsection discusses the prediction models to be experimented with in order to determine which college course is the most suitable for the user. As discussed in the previous subsection, the text that will be used from the user feedback is based on the highest sentiment score calculated. Thus, the words associated with the sentiment score will be used to determine the course. For this reason, this paper will be comparing two different models: K-Nearest Neighbours and Fuzzy Logic. Both models can predict a course by matching the course's keywords to the subjects in the user's response. The recommendations generated by the two models will be further evaluated based on their performance.

1) K-Nearest Neighbor: In K-Nearest Neighbors (KNN), the supervised machine learning algorithm acts as a text classification technique to categorize objects, in this case the college course. This is based on the text's similarity with other documents inside the corpus. KNN measures the likelihood of the object to be classified with the neighboring class located. In a course recommendation system, each course can be represented in a vector with its attributes consisting of keywords extracted using TF-IDF and N-gram previously. This representation of the course can be measured by numerical values based on the importance and relevance of the feature. The KNN model trains its data by first vectorizing the features. In this case, the college keywords and the college course's name would be considered. The model splits the student feedback data into train and test sets, whereby the train set will be used to perform and develop the prediction algorithm. Once the data is trained, the system can undergo finding the nearest neighbors for each test sample. Setting the k value 6, the model would be considering 5 nearest courses from the user response to retrieve. From this, the KNN model can generate 5 college courses that are found most matched to the description the user is referring to. These suggested courses will be compared to the ground truth with the hope that the actual course is at least in the top 5 recommended courses for the user.

2) Fuzzy Logic: Fuzzy Logic, a mathematical approach, categorizes or represents information with partial truth. Unlike binary Logic, where statements are either true or false, Fuzzy Logic allows for a degree of truth by assigning values between 0 and 1. With such flexibility, applications such as recommendation systems are highly suited to be performed using Fuzzy Logic. In Fuzzy Logic, sets of rules are set and the results are described in natural language terms. Fuzzy Logic then maps these linguistic values to numerical values to facilitate mathematical operations. In this college course recommendation system, Fuzzy Logic uses text similarity computation and a ranking approach to recommend a suitable course based on user feedback. Similarity scores in the course keywords can be calculated using the Fuzzywuzzy library to determine the association between the user's feedback and the keywords related with each course. Next, the computed similarity scores are paired with the corresponding course names which are sorted based on the scores. The top 5 courses with the highest fuzzy matching scores are selected as the predicted most suitable courses for the user. The accuracy of the course is evaluated by comparing the predicted top courses with the actual course predicted for the user. Prediction accuracy is calculated by determining if the course is in the top recommended courses.

III. Results and Discussion

This section will thoroughly discuss the calculations and findings from the research. The objective of this section is to compare different technique's performances in order to finally conclude the best approach to produce a college course recommendation system. Starting with the main prediction models, the KNN and Fuzzy Logic models are used to overall predict the final college course using the texts provided by the user. The accuracy of the prediction is based on whether the actual predicted course is found inside the top courses recommended by the different models.

A. K-Nearest Neighbor

K-Nearest Neighbor classifies the prediction by neighboring class location. While it is proven that KNN can match suitable courses from the keywords found inside the user's response, it is worth noting that KNN's performance relies heavily on the feature extraction and the amount of training sets available. Thus, attempting to improve the accuracy of KNN might need to review the overall process. Figure. 5 displays the courses predicted from the test set based on the user's text. As seen in the graph below, the courses predicted by the model are relatively in the same educational field. For example, 'Electronic Engineering', 'Mechanical Engineering', 'Electrical Engineering' are courses that can be found in the field of 'Engineering'. This highlights the concept that the recommendations KNN provides are relevant to the user's feedback.

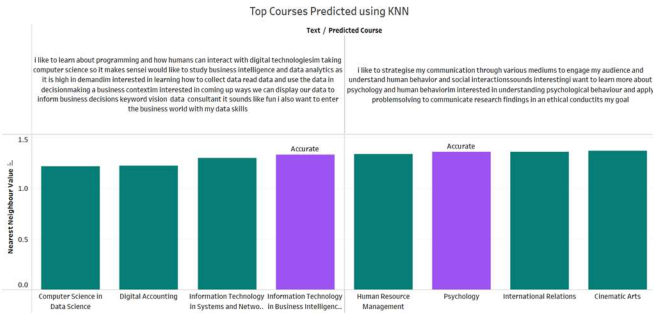

Fig. 5 Top Courses Predicted using KNN

B. Fuzzy Logic

Unlike the KNN model, Fuzzy Logic generates an accurate prediction based on the fuzzy rule set. The fuzzy Logic approach assigns a match score to the courses based on similarity. This fuzzy rule set allows a more personalized and adaptive recommendation system that considers the varying levels of relevance for different courses. In addition, Fuzzy Logic is well-suited for dealing with uncertainty and imprecision in data, which is often present in natural language processing tasks. Fuzzy Logic rule is more straightforward compared to KNN model as it only measures the similarity score between the course and the words extracted from the user's responses. Table 5 shows the top courses predicted using Fuzzy Logic.

C. Findings

As mentioned, the pre-processing and feature extraction methods greatly influence the prediction result. Table 6 displays the accuracy result retrieved from experimenting with different feature extraction and prediction model combinations. The highest results obtained are by using Fuzzy Logic, combined with a hybrid feature extraction algorithm, TF-IDF and N-gram. The prediction this approach received was 85.71%. Several factors may cause other approaches to be lower, such as the insufficiency of the training data set. However, Fuzzy Logic appears to adapt to dataset size better than KNN. Although there is not enough training and test model to verify the authenticity of both the models' accuracies, this working hypothesis can be used to set a benchmark for future research.

TABLE V

TOP COURSES PREDICTED USING FUZZY LOGIC

ID	Final Choice	Final Actual Course	Top Course	Prediction Accuracy
16	i like to learn about programming and how …	Information Technology in Business Intelligence	[Robotic Design, Computer Science in Game Dev …	True
6	i like to learn about organisations and unders ...	Business Management	[International Business, Entrepreneurship, Fin…	False
13	i like to strategize my communication …	Psychology	[Electrical Engineering, Electronic Engineering…	True
5	communication. I like to apply my mathema …	Digital Advertising	[Electronic Engineering, Entrepreneurship, Dig …	True
19	multimedia. I want to be an animator …	Animation	[Electronic Engineering, Robotic Design, Anima …	True
10	because it's okay the editing I guess in interes..	Visual Effects	[Entrepreneurship, Animation, Cinematic Arts …	True
17	Information technology computer science like ..	Information Technology in Intelligent System	[Information Technology in System and Network ...	True

TABLE VI

ACCURACY PERFORMANCE COMPARISON

Algorithms	Accuracy
K-Nearest Neighbor	57.14%
K-Nearest Neighbor + TF-IDF	71.43%
K-Nearest Neighbor + TF-IDF + N-gram	57.14%
Fuzzy Logic	57.14%
Fuzzy Logic + TF-IDF	71.43%
Fuzzy Logic + TF-IDF + N-gram	85.71%

In general, both prediction models manage to provide suitable choices, of course, mainly due to the structure of the data provided as well as the pre-processing of the data. One of the main issues the system is pursuing to tackle is measuring more qualitative data to predict the student's choices. In many existing recommendation systems, the amount of data to predict a personal recommendation for the user is limited as the input is mainly based on ratings and numeric values. However, this system provides a series of subjective questions for the user to answer to provide the system with a better choice of keywords to describe their interests in a college course. This allows predictive algorithms such as KNN to find the nearest college neighbor based on the user's texts. With the issue of the abundance of choices offered in different colleges, the process of this system's data collection can help the user narrow down the options based on the options they have picked. From the user's responses, the keywords they provided become more specific near the end of the questionnaires, which helps the sentiment analysis determine which course they enjoy more. The results at the end of both KNN and Fuzzy Logic also generated 5 top choices that the system found most appropriate for the user, which solves the broad and vague options the user faced in the beginning.

IV. CONCLUSION

This study explores various methods for developing a college course recommendation system, focusing on incorporating a sentimental approach to gather data and enhance the accuracy of recommendations. However, the research faces challenges due to limited student data, resulting in the cold start issue. Ongoing data collection efforts are expected to improve the training data and enhance the performance of predictive models. Another challenge lies in recommending precise college courses to students. With numerous course options and similar descriptions, the system may recommend courses that are not the students' primary preferences. To address this, the research aims to refine the feature extraction process for the college course and student feedback datasets. Techniques such as Latent Dirichlet Allocation (LDA) will be employed to uncover underlying topics within student responses.

Furthermore, future planning involves developing a user-friendly interface that allows students to provide feedback and receive instant course suggestions. The collected responses will expand the available data and improve the recommendation system. In conclusion, this study strives to advance college course recommendation systems by conducting comprehensive research, addressing challenges such as limited data and imprecise recommendations, and enhancing the user interface for a seamless experience.

REFERENCES

[1] Y.-K. Ng and J. Linn, "CrsRecs: A personalized course recommendation system for college students," in *2017 8th International Conference on Information, Intelligence, Systems & Applications (IISA)*, IEEE, Aug. 2017, pp. 1–6. doi: 10.1109/IISA.2017.8316368.

[2] M. J. Gomez, M. Calderón, V. Sánchez, F. J. G. Clemente, and J. A. Ruipérez-Valiente, "Large scale analysis of open MOOC reviews to support learners' course selection," *Expert Syst Appl*, vol. 210, p. 118400, Dec. 2022, doi: 10.1016/j.eswa.2022.118400.

[3] R. Kumar and S. Bhatia, "Course Recommender System Architecture with Sentiment Score," *SSRN Electronic Journal*, 2021, doi: 10.2139/ssrn.3879643.

[4] A. Esteban, A. Zafra, and C. Romero, "Helping university students to choose elective courses by using a hybrid multi-criteria recommendation system with genetic optimization," *Knowl Based Syst*, vol. 194, p. 105385, Apr. 2020, doi: 10.1016/j.knosys.2019.105385.

[5] J. Li and Z. Ye, "Course Recommendations in Online Education Based on Collaborative Filtering Recommendation Algorithm," *Complexity*, vol. 2020, pp. 1–10, Dec. 2020, doi: 10.1155/2020/6619249.

[6] R. Morsomme and S. Vazquez Alferez, "Content-based Course Recommender System for Liberal Arts Education," in *Proceedings of The 12th International Conference on Educational Data Mining (EDM 2019)*, 2019, pp. 748–753.

[7] H. Ezaldeen, R. Misra, S. K. Bisoy, R. Alatrash, and R. Priyadarshini, "A hybrid E-learning recommendation integrating adaptive profiling and sentiment analysis," *Journal of Web Semantics*, vol. 72, p. 100700, Apr. 2022, doi: 10.1016/j.websem.2021.100700.

[8] Z. Gulzar and A. A. Leema, "Course Recommendation Based on Query Classification Approach," *International Journal of Web-Based Learning and Teaching Technologies*, vol. 13, no. 3, pp. 69–83, Jul. 2018, doi: 10.4018/IJWLTT.2018070105.

[9] V. N. and A. K. K.M., "E-learning course recommendation based on sentiment analysis using hybrid Elman similarity," *Knowl Based Syst*, vol. 259, p. 110086, Jan. 2023, doi: 10.1016/j.knosys.2022.110086.

[10] J. Mawane, A. Naji, and M. Ramdani, "Recommender e-learning platform using sentiment analysis aggregation," in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Sep. 2020, pp. 133–138. doi: 10.1145/3419604.3419784.

[11] X. Jiang, L. Bai, X. Yan, and Y. Wang, "LDA-based online intelligent courses recommendation system," *Evol Intell*, Oct. 2022, doi: 10.1007/s12065-022-00810-2.

[12] Z. Hasan and S. S. Baskaran, "Propose a Recommender System to Dynamically Align Higher Education Curriculums With 4IR Market Needs," in *2023 International Conference on IT Innovation and Knowledge Discovery, ITIKD 2023*, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/ITIKD56332.2023.10099924.

[13] M. R. Julianti, Y. Heryadi, B. Yulianto, and W. Budiharto, "Recommendation System Model for Personalized Learning in Higher Education using Content-Based Filtering Method," in *2022 International Conference on Information Management and Technology (ICIMTech)*, IEEE, Aug. 2022, pp. 1–6. doi: 10.1109/ICIMTech55957.2022.9915109.

[14] H. Ng, M. S. Jalani, T. T. V. Yap, and V. T. Goh, "Performance of Sentiment Classification on Tweets of Clothing Brands," *Journal of Informatics and Web Engineering*, vol. 1, no. 1, pp. 16–22, Mar. 2022, doi: 10.33093/jiwe.2022.1.1.2.

[15] Z. Shahbazi and Y.-C. Byun, "Agent-Based Recommendation in E-Learning Environment Using Knowledge Discovery and Machine Learning Approaches," *Mathematics*, vol. 10, no. 7, p. 1192, Apr. 2022, doi: 10.3390/math10071192.

[16] V. Vanitha, P. Krishnan, and R. Elakkiya, "Collaborative optimization algorithm for learning path construction in E-learning," *Computers & Electrical Engineering*, vol. 77, pp. 325–338, Jul. 2019, doi: 10.1016/j.compeleceng.2019.06.016.

[17] J. S. V. Leong and K. B. Gan, "Cuffless Non-invasive Blood Pressure Measurement Using CNN-LSTM Model: A Correlation Study," *International Journal on Robotics, Automation and Sciences*, vol. 5, no. 2, pp. 25–32, Sep. 2023, doi: 10.33093/ijoras.2023.5.2.3.

[18] R. Alatrash, H. Ezaldeen, R. Misra, and R. Priyadarshini, "Sentiment Analysis Using Deep Learning for Recommendation in E-Learning Domain," in *Advances in Intelligent Systems and Computing*, Springer Science and Business Media Deutschland GmbH, 2021, pp. 123–133. doi: 10.1007/978-981-33-4299-6_10.

[19] T. Kumar *et al.*, "Fuzzy Logic and Machine Learning-Enabled Recommendation System to Predict Suitable Academic Program for Students," *Math Probl Eng*, vol. 2022, pp. 1–7, Aug. 2022, doi: 10.1155/2022/5298468.

[20] S. Ghosh, "A Hybrid Programming Course recommendation system using Fuzzy Logic and xDeepFM," in *2023 International Conference on Intelligent Systems, Advanced Computing and Communication (ISACC)*, IEEE, Feb. 2023, pp. 1–8. doi: 10.1109/ISACC56298.2023.10083853.

[21] S. A. Stein, G. M. Weiss, Y. Chen, and D. D. Leeds, "A College Major Recommendation System," in *RecSys 2020 - 14th ACM Conference on Recommender Systems*, Association for Computing Machinery, Inc, Sep. 2020, pp. 640–644. doi: 10.1145/3383313.3418488.

[22] J.-P. Cheng and S.-C. Haw, "Mental Health Problems Prediction Using Machine Learning Techniques," *International Journal on Robotics, Automation and Sciences*, vol. 5, no. 2, pp. 59–72, Sep. 2023, doi: 10.33093/ijoras.2023.5.2.7.

[23] S. T. Lim, J. Y. Yuan, K. W. Khaw, and X. Chew, "Predicting Travel Insurance Purchases in an Insurance Firm through Machine Learning Methods after COVID-19," *Journal of Informatics and Web Engineering*, vol. 2, no. 2, pp. 43–58, Sep. 2023, doi: 10.33093/jiwe.2023.2.2.4.

[24] N. P, K. Saiteja, K. K. Ram, K. M. Kanta, S. K. Aditya, and M. V, "University Recommender System based on Student Profile using Feature Weighted Algorithm and KNN," in *2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)*, IEEE, Apr. 2022, pp. 479–484. doi: 10.1109/ICSCDS53736.2022.9760852.

[25] K. K. Jena *et al.*, "E-Learning Course Recommender System Using Collaborative Filtering Models," *Electronics (Basel)*, vol. 12, no. 1, p. 157, Dec. 2022, doi: 10.3390/electronics12010157.

[26] H. El Fazazi, M. Qbadou, I. Salhi, and K. Mansouri, "Personalized recommender system for e-Learning environment based on student's preferences," in *IJCSNS International Journal of Computer Science and Network Security*, 2018.

[27] H. Tang *et al.*, "Predicting Green Consumption Behaviors of Students Using Efficient Firefly Grey Wolf-Assisted K-Nearest Neighbor Classifiers," *IEEE Access*, vol. 8, pp. 35546–35562, 2020, doi: 10.1109/ACCESS.2020.2973763.

[28] A. Odry, V. L. Tadic, and P. Odry, "A Stochastic Logic-Based Fuzzy Logic Controller: First Experimental Results of a Novel Architecture," *IEEE Access*, vol. 9, pp. 29895–29920, 2021, doi: 10.1109/ACCESS.2021.3059482.

[29] A. A. Zaveri, R. Mashood, S. Shehmir, M. Parveen, N. Sami, and M. Nazar, "AIRA: An Intelligent Recommendation Agent Application for Movies," *Journal of Informatics and Web Engineering*, vol. 2, no. 2, pp. 72–89, Sep. 2023, doi: 10.33093/jiwe.2023.2.2.6.

[30] T. R. Razak, J. M. Garibaldi, C. Wagner, A. Pourabdollah, and D. Soria, "Toward a Framework for Capturing Interpretability of Hierarchical Fuzzy Systems—A Participatory Design Approach," *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 5, pp. 1160–1172, May 2021, doi: 10.1109/TFUZZ.2020.2969901.

[31] M. K. Sharma, N. Dhiman, Vandana, and V. N. Mishra, "Mediative fuzzy logic mathematical model: A contradictory management prediction in COVID-19 pandemic," *Appl Soft Comput*, vol. 105, p. 107285, Jul. 2021, doi: 10.1016/j.asoc.2021.107285.