Professor Recommendation System

1st Madhu Ashala
Department of Information Science
University of North Texas
Texas, United States
madhuashala@my.unt.edu

2nd Kamini Suresh Yelamar
Department of Information Science
University of North Texas
Texas, United States
kaminisureshyelamar@my.unt.edu

3rd Rangineni Rakshith Rao
Department of Information Science
University of North Texas
Texas, United States
rakshithraorangineni@my.unt.edu

4th Aakanksha Srivastava
Department of Information Science
University of North Texas
Texas, United States
aakankshasrivastava@my.unt.edu

***Abstract*—Choosing the right professor is an important decision which every student has to take while enrolling for a course and which would impact their success in the course as well as their academic experience. The information about professor's research interest, teaching style, course difficulty level etc is not readily available and is mostly unstructured making it difficult for students to find the right professor. Our study helps resolve this issue by developing a professor recommendation system which would recommend professor based on students research interest, academic goal and learning preference. Course feedback and student reviews are also taken into consideration while providing tailored recommendations. Various data science techniques such as web scraping, data preprocessing and Natural Language Processing (NLP) have been used to extract data and analyze textual data from student feedback and publications simultaneously. By integrating various data sources and applying algorithms such as TF-IDF and Cosine similarity, a personalized professor recommendation system has been developed which would align with each students unique requirements. This project provides UNT students with a very powerful tool which would help them make informed decision while choosing professor and enhance their academic experience.**

***Index Terms*—Term Frequency - Inverse Document Frequency (TF-IDF), Cosine Similarity, Natural Language Processing (NLP), K-Nearest Neighbors**

I. Introduction And Problem Statement

Professor plays a crucial role in student's academic journey, as it directly affects their academic success and learning experience. But finding the right one can be difficult because information about teachers, such as their teaching styles, research interests, and levels of learning difficulty, is often scattered or unstructured. Students often rely on word of mouth recommendations or on imperfect materials, which may not always be the case for the best results. Research shows that matching a student's learning preferences with a professor's teaching style can significantly enhance their learning experience and performance [1].

To address this problem, we developed a Professor Recommendation system that uses state-of-the-art data science techniques to provide relevant recommendations to students. The system integrates information from sources such as professor publications, course notes, and students' SPOT evaluation to match student research interests, learning objectives, and preferred learning styles Tools such as web scraping, data preprocessing, and natural language processing. Using (NLP) , the system efficiently analyzes structured and unstructured data [2].

Large-scale designs such as TF-IDF (Term Frequency-Inverse Document Frequency) and Cosine Similarity have been used to assess teachers' fit to student input [3][4]. These methods are well known for their effectiveness in textual - based recommendation systems. Our system provides UNT students with a powerful tool to make informed decisions when choosing faculty, helping them determine the best fit for their learning needs and enhancing their overall learning experience. By combining multiple data sources with personalized recommendations, the system bridges the gap between students and teachers, and ensures a good fit that leads to learning outcomes [6].

This project demonstrates how advanced data science methods can solve real-world problems in education. Through a mixture of structured and unstructured data analysis, our work highlights the value of recommendation systems in facilitating the decision-making process for students. In the sections that follow, we will detail the methods we used, the challenges we faced, the results we obtained, and the overall impact of our work, and show how it improves faculty-student alignment and fosters learning the experience is great.

II. Literature Review

Personalized course recommendation systems
The system DORIS was introduced as referenced in [12], which personalized recommendations for courses and learned about the student profiles, behavioral patterns, and past academic performance. Findings suggest that deep neural networks have been proven most potent at identifying complex relationships between students and courses relative to previous

models. This work holds special relevance to our project, seeing as they show how advanced machine learning models are very good at providing personalized recommendations, which are also extendable to recommending professors[12].

The authors of [13] targeted its work in personalized online learning by constructing a system committing adaptive learning paths based on students' individual needs and learning styles. Basically, this system improves the quality of online education by creating efficient and effective learning environments. Their approach is of considerable significance to our work, as it stresses the necessity to customize recommendations according to individual learning preferences a goal excellently correlated with attempting to recommend professors by similar principles [13].

Professor Recommendation Systems

A professor recommendation system was developed in [14], which helps students choose professors based on research interests, teaching methods, and course expertise. This work closely resembles our work in many ways and aims to create a match between professors and students in meeting the former's academic goals. This, in turn, improves educational outcomes and student satisfaction [14].

Ontology and Semantic Techniques in Recommendations

Authors of the research in [15] proposed an ontology-based framework for helping provide personalized recommendations of courses to students. Using structured data and semantic reasoning, this course recommendation system is able to match students with courses most compatible with their interests. This changes the whole dynamics of recommendation system by making it more precise through disambiguation and strengthening user preference negotiations. The semantic techniques employed by them could be extended to provide a similar match of students with professors based on research interests and teaching methods [15].

Personalized Learning Resources

An algorithm proposed in [16] to recommend personalized learning resources, such as course materials and study aids that would suit the preferences of the student. It is considered that the personalized and student-centered systems employing data are effective for improving educational experiences. This also sums up one of the objectives of our project, wherein data should be utilized for professor recommendations [16].

Recommendation Systems in Learning Management Systems

An Integrating machine learning is proposed in [9] for Learning Management Systems to deliver personalized recommendations. Using collaborative and content filtering, their system matches educational resources with students' learning history, preferences, and behavior. Their methodology demonstrates the potential of machine learning in designing personalized teaching resources [9].

Movie Recommendation Systems

As discussed in [8] a movie recommendation system that is based on collaborative and content-based filtering. They addressed problems such as scalability and accuracy, suggesting neural networks and matrix factorization as solutions. While entertainment-centered in its considerations, the concepts introduced can be applied for educational recommendations, especially for challenges like data scalability and accuracy [8].

Algorithms in Recommendation Systems

Several algorithms using machine learning like k-nearest neighbors, support vector machines, and decision trees have been applied to recommendation systems in [7] by authors. They demonstrated that each algorithm has strengths and weaknesses in different contexts such as e-commerce and online advertising. This study also highlighted the importance of choosing the right algorithms in order to satisfy the user and run the system correctly [7].

Automated Resume Recommendation

A machine learning-based system developed in [10] for resume recommendations by using natural language processing to extract key features and providing rank based on decision tree and support vector machine models for recommending candidates. The system is quite capable of matching skills with organizational needs, thus proving a platform to implement similar measures for matching students with professors [10].

Hotel Recommendation Systems

The research [11] focused on recommending hotels tailored to customer requirements as a way to tell which algorithm or methodology to select from. The study explored machine learning algorithms such as neural networks, decision trees, and support vector machines as common methods to boost the accuracy and relevance of recommendations. This forms the basis on which algorithm selection might be performed in similar contexts[11].

III. Objectives of the Study

1) To create a machine learning model which recommends professors depending on various factors such as course preferences, difficulty level, feedback from past students and research interests to ensure personalized suggestions.
2) Provide accurate and relevant recommendations using advanced NLP techniques by studying semi-structured data which includes research areas, publications, student feedback.
3) Establish a link between professors' research expertise, teaching style, class engagement and student's learning preferences to tailor suggestions based on career and academic goals of an individual.

4) To develop a user-friendly interface where students can enter their preferences and get recommendations accordingly.

IV. Data Collection

The data to be used in this study was acquired from publicly accessible resources from the University of North Texas web sites using the Python web scraping tool and the Selenium web scraping framework. Faculty background information, including name, title, courses taught, research publications, department, spot related data such as the Challenge and Engagement Index, Overall Summative Rating, number of enrolled students, and number of responses to course evaluation is retrieved from the UNT Faculty Information portal. The data was valuable in informing student engagement and teaching performance for different courses.

Additional faculty research areas were found on the UNT Experts portal. Since this content is generated dynamically by JavaScript, Selenium was used to navigate the site, and Beautiful Soup for processing and structuring data. Using this method, the information collected is complete and accurate enough to rely on in the analysis process. The data sources are the UNT Faculty Information portal: https://facultyinfo.unt.edu/ and the UNT Experts portal: https://experts.unt.edu/. And the final pre-processed dataset can be found at : InfoData.

V. Data Processing and Analysis

A. Data Pre-processing :

Data processing aims to clean and improve our dataset. Two datasets were created first which focuses on professor specific keywords/expertise and the other which had details of name, title, department, courses, semesters and Student Perspective Of Teaching (SPOT evaluation) results. Then these were combined using profile ID and a comprehensive dataset was created which had all the required information. After merging, the dataset on Profile ID of faculty, it was carefully refactored to ensure consistency in the datatype and its structure. The first version of dataset had below columns:

- **Content**:
  - **Faculty Name**: The full name of the professor.
  - **Title**: The academic title or position of the professor.
  - **Department**: The department to which the professor belongs.
  - **College**: The college or school the professor is affiliated with.
  - **Course Code**: The specific course the professor is teaching.
  - **Course Name**: The name of the course.
  - **Semester**: The semester in which the course is offered.
  - **Employee ID (EID)**: A unique identifier for the professor.
  - **Overall Summative Rating**: A rating based on student feedback, representing overall teaching effectiveness.
  - **Number of Responses**: The number of students who responded to the feedback survey.
  - **Number of Students**: The total number of students enrolled in the course.
  - **Challenge and Engagement Index**: A measure of how challenging the course was and the level of student engagement.
  - **Publications/Research Interests**: A list of the professor's publications and research focus areas.

Fig. 1. Data Description

The raw data scraped from UNT websites had 7077 rows and 12 columns. Approximately 4330 records were having missing values in Overall Summative Rating, Challenge and Engagement Index, Number of Responses/ Students, Publications and keywords columns. A systematic approach was then followed to address these gaps and improve the dataset. To handle this missing at random (MAR) data in numerical columns such as Overall Summative Rating, Challenge and Engagement Index, Number of Responses/ Students mean imputation has been performed for respective faculty. This helps us in making sure that our data retains its context while filling gaps. And the textual columns (Missing Not at Random - MNAR) e.g., publications and keywords were imputed by manually browsing about faculty on publicly available websites such as Google Scholar, LinkedIn and their CVs. At the end for the faculty where no data was found (Missing Completely at Random- MCAR) we inserted values as 'No Keywords/Publications Found' in the dataset. By doing this the amount of missing data was significantly reduced and helped us get a more reliable dataset.

After handling the missing data systematically, we performed cleaning and mining tasks on the textual columns. Few of the tasks included replacing irrelevant text with empty strings, removing stop-words, punctuations, numbers and regularizing the datatype of columns. Followed by removing current and future semesters' records, since it would affect the system's performance because of unavailability of SPOT results/values.

B. Feature Engineering :

After obtaining the cleaned dataset, new metrics, Student Feedback Index (SFI) and Professor Engagement Index (PEI) were created which would help in deeper analysis. The new metrics were derived from the existing metrics like Challenge and Engagement Index, Number of Students, Number of Responses and Overall Summative Rating.

Course Difficulty Index (CDI) : This index aims to quantify the perceived difficulty of a course based on student engagement, response rates, and the challenge index provided. It is found by adding the three metrics Challenge and Engagement Index which is scaled to 60, Response Rate which is scaled to 20 and Overall Summative Rating which is also scaled to 20. The resulting score which we get after addition shows how difficult and challenging the course is based on feedback provided by students. The final score we get is then rounded to 2 decimal places.

The formula to calculate the CDI is :

$$\mathrm{CDI} = \left(\frac{\mathrm{CEI}}{7} \times 60\right) + \left(\frac{\mathrm{RR}}{100} \times 20\right) + \left(\frac{1}{\mathrm{OSR}} \times 20\right)$$

Where,

- CEI : Challenge and Engagement Index
- RR: Response Rate
- OSR: Overall Summative Rating

Difficulty Category : The Difficulty Category is a classification based on the calculated Course Difficulty Index.

The index evaluates the relative challenge level of a course based on various contributing factors. The classification is determined as follows:

- If the Course Difficulty Index is less than 70, the course is categorized as Easy.
- If the Course Difficulty Index is 70 or higher, the course is categorized as Hard.

This classification provides a straightforward way to understand and compare the difficulty levels of different courses.

Student Feedback Index (SFI) : It is a metric to measure the quality and representativeness of any given course. This evaluates the overall influence of a professor on student learning and engagement. The two main components Overall Summative Rating and Challenge and Engagement Index are multiplied to balance. Multiplication means that all parts depend on each other. If one part is weak, it pulls the whole score down. This makes sense for SFI because it's meant to show the **overall impact**, and strong performance is needed for a high score in all areas.

We further refine these values by accounting for the ratio of students who submitted feedback to the total number of students enrolled in the course. The SFI not only reflects the sentiment of the feedback but also incorporates the volume of responses to ensure a more representative evaluation. Finally, the calculated index is rounded to two decimal places for clarity and ease of interpretation.

The formula for the SFI is given as:

$$\text{SFI} = \text{OSR} \times \text{CEI} \times \left(\frac{\text{Number of Responses}}{\text{Number of Students}}\right)$$

Where,

- CEI : Challenge and Engagement Index
- OSR: Overall Summative Rating

Higher SFI scores indicate positive student feedback and satisfaction with the professor's teaching style and course management. The index helps students make informed decisions about selecting professors that align with their learning preferences and academic goals.

Professor Engagement Index (PEI) : This metric combines two critical components: the **Overall Summative Rating** and the **Challenge and Engagement Index**. These metrics are added together to provide an overall measure of course engagement and quality. Addition means each part adds to the score separately. If one part is weak, the other parts can still keep the score high. This is good for PEI because it's about **overall engagement**, and a professor can still be engaging even if one area isn't perfect.

To further refine the metric, the sum is adjusted by the proportion of students who provided feedback relative to the total number of students enrolled in the course. The resulting value is then rounded to two decimal places for better readability.

	Course Difficulty Index	SFI	PEI
count	6726.000000	6726.000000	6726.000000
mean	62.001481	12.273165	5.013061
std	6.121489	4.910132	1.815629
min	31.100000	0.520000	0.310000
25%	59.180000	9.070000	3.850000
50%	61.350000	11.150000	4.400000
75%	64.660000	13.900000	5.790000
max	84.400000	33.000000	11.600000

Fig. 2. Descriptive Statistics

$$\text{PEI} = (\text{OSR} + \text{CEI}) \times \left(\frac{\text{Number of Responses}}{\text{Number of Students}}\right)$$

Where,

- CEI : Challenge and Engagement Index
- OSR: Overall Summative Rating

A higher PEI indicates stronger student feedback and greater professor engagement, reflecting a more impaction and interactive teaching experience. These metrics provide a better picture about engagement and effectiveness of the professor. Unnecessary columns were removed from the dataset to make it easier to work with and in a more focused manner.

C. Exploratory Data Analysis

After exporting the pre-processed and feature engineered data to an excel, we performed an Exploratory Data Analysis on our final dataset to gain an initial understanding of the dataset and uncover potential patterns, trends, or irregularities. We started with studying the data types, size (6726*10) and descriptive statistics of numerical columns. The dataset had 110 unique professors' records for our research.

Performing uni-variate analysis helped us understand the data better. Bar charts were created for visualizing the distribution of different categorical variables like department codes and professor titles. We proceeded with checking the number of faculty members under each title and found that most of them had their title as 'Professor', followed by 'Adjunct Faculty', 'Principal Lecturer', 'Chair' and so on. (See Fig 3)

UNT College of Information offers a total of 294 courses. While examining the number of courses offered under each department code, we found that the majority of courses were from the INFO department, followed by the SLIS and ATTD departments. On the other hand, the PSYC, BCIS, and BAAS departments had the fewest offerings, with just one course each. (See Fig 4)

The distribution for CDI, SFI and PEI has been plotted to understand the spread and measure of dispersion. The analysis indicates that most values for SFI cluster between approximately 10 and 15, PEI values between 4 and 6 and CDI values are centered around 60, with most values falling between 55 and 65. (See Fig 5)

Analyzing the keywords provided deeper insights into the dataset. Keywords associated with each professor were refined by removing multi-word phrases and common terms. A word cloud was then created, highlighting the most frequently mentioned words, making it easier to identify key terms at a glance. (See Fig 6)

VI. Research Hypothesis

The research focuses on the development of a recommendation system that uses a classification model and user interface, based on UNT faculty information and student preferences. This approach will significantly improve the reliability, relevance of course and professor recommendations for UNT students. Our focus is to not only enhance their academic experience but also satisfaction.

The hypothesis states that integrating metrics such as Student Feedback, Professor Engagement and Course Difficulty with the research interests will improve the reliability of Professor Recommendations, leading to better alignment between student preferences and academic outcomes.

VII. Data Visualization

The bar chart in Fig 3 shows the counts of different titles in the dataset.

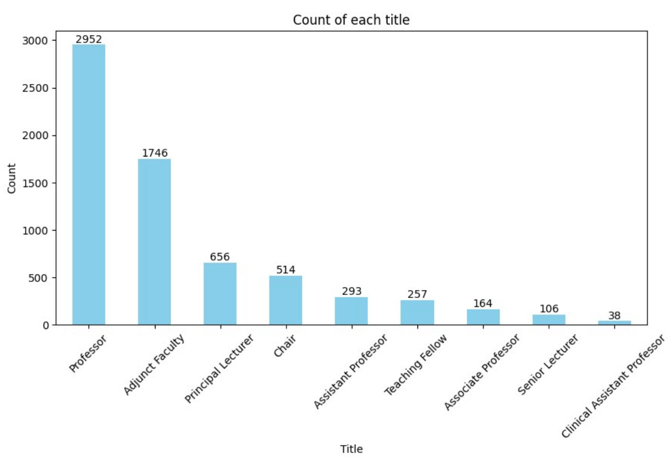

Fig. 3. Titles Vs Count

The bar chart in Fig 4 displays department wise count, showing Information Science as a dominant department.

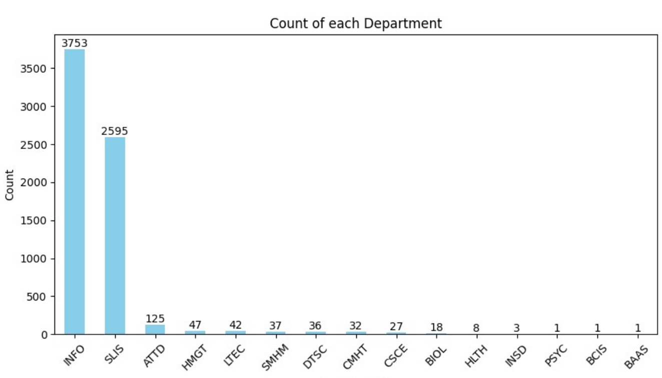

Fig. 4. Department Vs Count

The histograms in Fig 5 shows different frequency distributions for SFI, PEI and CDI.

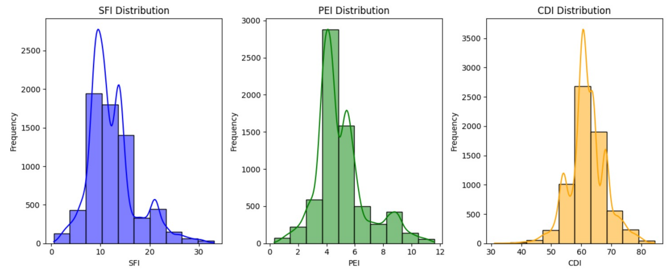

Fig. 5. Frequency Distribution of Metrics

Below word cloud in Fig 6 has been created for keywords of one of the faculty.

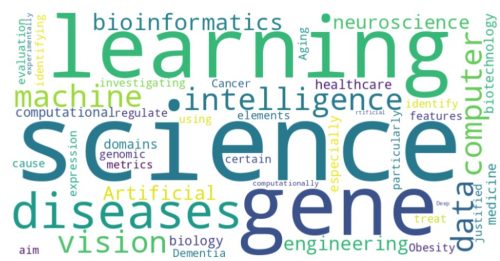

Fig. 6. Word Cloud

VIII. Implementation

The professor recommendation system has been designed to facilitate the students at University of North Texas, it helps students to make insightful decisions while selecting professors. It is a web application which has been built based on Flask and integrates Recommendation Algorithms, NLP techniques, data preprocessing to deliver personalized suggestions. The recommendation system takes input such as Course codes, Academic Interest, and difficulty levels to suggest professors aligned with student preferences, making sure that the suggestions made are connected and effective.

The base of the system lies in its data preparation pipeline, which converts the raw web-scraped input data into a well-structured, processable format. Data has been fetched from the UNT website from faculty profiles and course information, including key fields such as Course details, professor names, research interests, and feedback metrics. The steps involved in pre-processing stage are data cleaning and mining by removing stop words, punctuation, and irrelevant characters using Natural Language processing libraries.

Further to improve the system's precision, new features were created to study the professor efficacy. One of the features is Combined features column which brings the data together such as course name, research interests, and related keywords. This provides a complete overview of each professor's profile. In addition, other features like the Professor Engagement Index (PEI), Student Feedback Index (SFI) and Course Difficulty Index(CDI) were introduced.

The **TF-IDF** (Term Frequency-Inverse Document Frequency) is considered as the heart of the system which turns the text into numbers. This method identifies the significant terms in a faculty profile and helps the system to focus more on what really matters. The inputs by the students such as course name and academic interests are converted into vector and can be compared with the professor's profile. The **Cosine Similarity** is then used to determine how closely a professor's profile matches the student's input, resulting in a similarity score that is used to rank the recommendations. It ensures that the professor is recommended only once without any repetitions and only top five professors are shown along with their SFI and PEI scores.

After getting a cosine similarity value, we are focusing on our numerical variables. To generate a reliable and robust recommendations, along with the textual columns we considered and used numerical values. For our research we have chosen K Nearest Neighbors algorithm to make the most suitable recommendation. The number of neighbors has been decided after performing cross validations with different values and their results. Final value for the number of neighbors is 5 with the metric as euclidean. The features given to the K-Nearest Neighbors (KNN) model includes Cosine similarity calculated for given Student input, Course Difficulty Index, Student Feedback Index and Professor Engagement Index. Using these metrics KNN model will return the most suitable Professor based on euclidean distance. The complete project flow can be drawn as below in Fig 7:

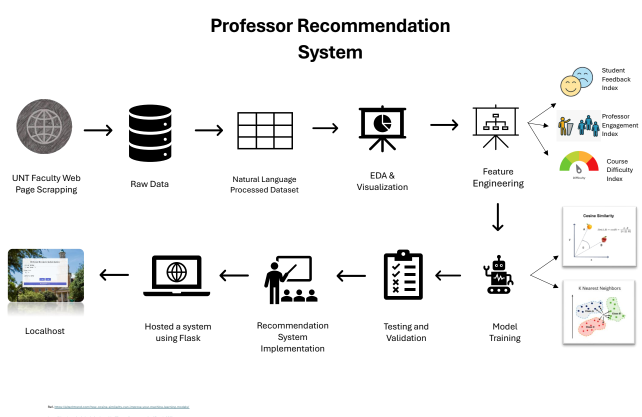

Fig. 7. Project Flow

The user interface of the system is designed using Flask in such a way that it is very simple and easy to use . Students can give the inputs according to their preferences and submit the form. The back-end process gets triggered and generates a list of recommended professors. This interaction has been optimized for speed and accessibility, allowing students to make informed decisions fast. The UI is shown in Fig 8:

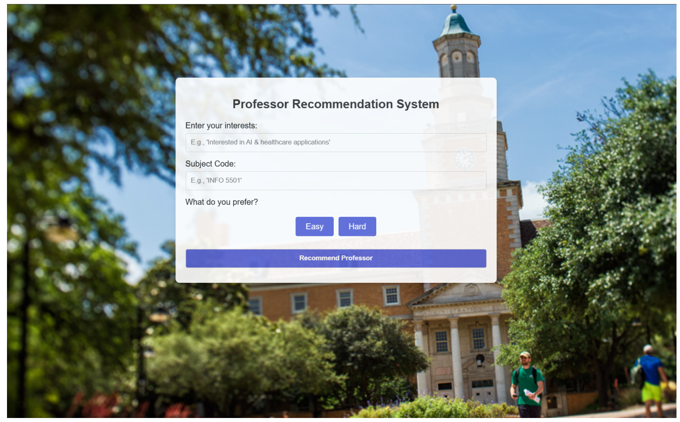

Fig. 8. User Interface

To improve reliability, the system integrates robust error handling. Few of the test cases as below:
1. The fields are made required and if a student enters wrong or incomplete input, the system validates the input and provides clear feedback so they can correct it. By doing this, the system becomes more reliable and ensures that users get the best possible recommendations.

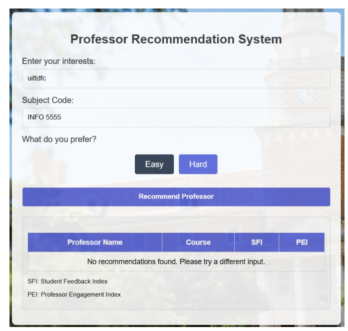

Fig. 9. Test1

2. The difficulty level can be chosen from UI, and system will give you results accordingly. For instance, lets test the recommendations for same input but with different difficulty levels:

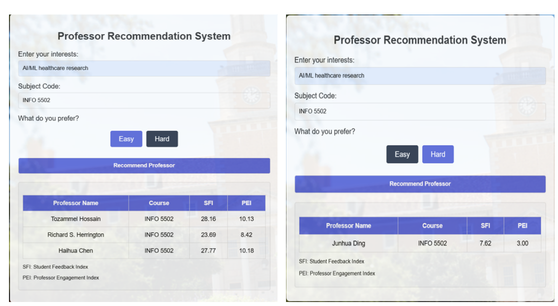

Fig. 10. Test 2

To summarize, the professor recommends system generates tailored recommendations by using advanced techniques in data processing, natural language processing, and machine learning to provide personalized recommendations. Students at UNT can use the system to make well-informed judgments regarding their professors and courses by concentrating on both the academic fit and the quality of the instructors' instruction.

IX. Conclusion

Developing a professor recommendation system for University of North Texas marks a significant step towards improving the process of decision making for the students. Using advanced techniques like Natural Language Processing, Machine Learning and Data Preprocessing the professor recommendation system provides tailored and insightful suggestions based on student preferences. The dataset was cleaned and enriched by Exploratory Data analysis which enabled the creation of new metrics Student Feedback Index (SFI) and Professor Engagement Index (PEI) for better evaluation. Integrating TF-IDF, Cosine similarity, and KNN ensures that the recommendations provided are relevant, efficient and accurate. Deeper insights were provided by visualization which helped us understand patterns and trends better and further strengthened the system's capability. A reliable and user-friendly web application was developed which helps students make informed decisions and helps enhance their academic experience and satisfaction.

X. Future Scope

The Professor Recommendation System has multiple expansion ideas. One of the major idea could be the development of a one database that stores user input, feedback, and interaction data. This can serve as a feedback loop for the model, wherein it learns from the data over time and becomes more accurate and relevant in its recommendations. Also, normalizing some of the key metrics - CDI, PEI, and SFI-in a scale of 1 to 10 will go a long way in enhancing the interpret-ability of the outputs and therefore a more user-friendly result. Automation testing can also be implemented to examine the recommendations using prompt inputs and the result received. Another future work is to add feedback from the users where students rate the recommendations or provide qualitative input about their satisfaction with the suggested professors. This feedback can be very important for fine-tuning the model.

References

[1] Smith, J., & Jones, R. (2020). *The Impact of Learning Preferences on Academic Success*. Journal of Educational Research, 45(2), 123–135.

[2] Brown, K., Zhang, L., & Williams, M. (2019). *Applications of NLP in Education: A Survey*. Computational Linguistics Journal, 12(3), 101–115.

[3] Jones, K. S. (1972). *A Statistical Interpretation of Term Specificity and Its Application in Retrieval*. Journal of Documentation, 28(1), 11–21.

[4] Salton, G. (1983). *Introduction to Modern Information Retrieval*. McGraw-Hill.

[5] Zhang, T., Li, P., & Zhou, X. (2021). *Sentiment Analysis in Education: A Framework for Understanding Student Feedback*. Educational Data Science, 8(4), 299–317.

[6] Miller, A., & Clark, D. (2022). *Optimizing Professor-Student Matchmaking through Data Science*. Journal of Higher Education Analytics, 15(3), 205–222.

[7] Nawrocka, A., Kot, A., Nawrocki, M. (2018). Application of machine learning in recommendation systems. In 2018 23rd International Conference on Methods and Models in Automation and Robotics (MMAR) (pp. 327-331). IEEE.

[8] Furtado, F., Singh, A. (2020). Movie recommendation system using machine learning. International Journal of Research in Industrial Engineering, 9(1), 84–98.

[9] Murad, D. F., Heryadi, Y., Wijanarko, B. D., Isa, S. M., Budiharto, W. (2018). Recommendation system for smart LMS using machine learning: A literature review. In 2018 4th International Conference on Computing, Engineering, and Design (ICCED) (pp. 36-41). IEEE.

[10] Roy, P. K., Chowdhary, S. S., Bhatia, R. (2020). A machine learning approach for automation of resume recommendation system. In Proceedings of the International Conference on Computational Intelligence and Data Science (ICCIDS 2019). Procedia Computer Science, 167, 2317–2325.

[11] Ramzan, B., Bajwa, I. S., Jamil, N., Amin, R. U., Ramzan, S., Mirza, F., Sarwar, N. (2019). An intelligent data analysis for recommendation systems using machine learning. Scientific Programming, 2019, Article 5941096, 20 pages.

[12] Ma, Y., Ouyang, R., Long, X., Gao, Z., Lai, T., Fan, C. (2023). DORIS: Personalized course recommendation system based on deep learning. PLoS One, 18 (6), e0284687.

[13] Lahiassi, J., Aammou, S., Warraki, O. E. (2023). Enhancing personalized learning with a recommendation system in private online courses. Conhecimento Diversidade, 15(39), 176-189.

[14] Mohammad, U., Hamdan, Y., Sardesai, A., Gokgol, M. (2023). A Novel Algorithm for Professor Recommendation in Higher Education. London Journal of Social Sciences, (6), 12-19.

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

[16] Zhang, L., Zeng, X., Lv, P. (2022). Higher education-oriented recommendation algorithm for personalized learning resource. International Journal of Emerging Technologies in Learning (Online), 17(16), 4.