Amsterdam University College Course Recommender : Facilitating Academic Planning

Adriana Zarańska Amsterdam University College

Barbara Plebanowicz Amsterdam University College

Leon Wloch Amsterdam University College

1. INTRODUCTION

One of the hardest choices that students in liberal arts programmes face is choosing which courses to pick. As students at the Amsterdam University College (AUC) in a Liberal Arts and Sciences programme, we understand this struggle firsthand. For this reason we decided to create a recommender system to aid students in making sound academic choices.

1.1 Relevance to Text Mining

We have utilised multiple key aspects of text mining throughout: preprocessing and cleaning of data, feature extraction through vectorisation, and undertaking a similarity analysis

1.2 Background

In the last few decades, recommenders have become increasingly common. Most modern services such as social media, streaming platforms, and shopping platforms utilise recommender systems as part of their product. There are many choices to make when implementing a recommender; one must choose between collaborative or content-based filtering, as well as memory-based or model-based methods.

Collaborative filtering as described by Algarni & Sheldon (2023) “is a technique that can filter out items that a user might like on the basis of reactions by similar users (i.e., searching a large group of people and finding a smaller set of users with tastes similar to a particular user)” (p. 566).

Algarni & Sheldon (2023) also describe the content-based filtering by stating: “the basic goal of content-based recommenders is to provide recommendations for products based on how various users or goods are similar to one another” (p. 566). The recommender does this by determining the features that distinguish the user’s most highly rated goods (Algarni & Sheldon, 2023). This type of recommender “can identify the user’s individual preferences and can suggest uncommon items that are not of much interest to other users“ (Algarni & Sheldon, 2023, p.566).

Memory-based recommenders suggest “new items by taking into consideration the preferences of its neighbourhood” (Roy & Dutta, 2022, p. 4). These make “use of the utility matrix directly for prediction” (Roy & Dutta, 2022, p. 4).

Roy & Dutta (2022) states that model-based systems “do not rely on the complete dataset when recommendations are computed but extract features from the dataset to compute a model” (p. 5).

Therefore, these systems are called model-based (Roy & Dutta, 2022). They further describe that these tequinches need two steps to make predictions: “the first step is to build the model, and the second step is to predict ratings using a function (f) which takes the model defined in the first step and the user profile as input” (p. 5).

We were not the only ones who came up with the idea of helping our peers with their course selections, as there is a paper written by researchers from University College Maastricht who also made a course recommender for their Liberal Arts programme based on courses available in their programme. Morsomme & Alferez (2019) also developed a content-based course recommender. However, their recommender system outputs 20 course recommendations, with additional warnings when the program predicts failure in the given course and with those warnings suggests preparatory courses (Morsomme & Alferez, 2019). The recommender developed by our team, however, outputs only five courses and does not output such warnings as this is outside our scope. It does not collect any data from the user except for the courses that they have taken.

1.3 Author Contributions

Adrianna: Evaluation of the Results, Recommender Function, Background, Limitations & Future Research

Barbara: Preprocessing of the Data, SentBERT Model, Recommender Function, Introduction, Methods

Leon: Data Collection, TF-IDF Vectorisation, Evaluation of the Results, Overall Notebook & Report Quality

2. METHODS

2.1 Data Collection

We collected our data from the UvA course catalogue website. First, we scraped all course links using chrome browser web drivers from the Selenium Python library (Selenium Contributors, 2025). Next, we extracted relevant information from these websites using the Beautiful Soup 4 library (Richardson, n.d.) to parse their HTML.

2.2 Data Description

The dataset obtained from the web scraping consisted of 3812 courses, but after removing courses not taught in English that number was reduced to 3345 courses. This data was stored in a Pandas (The pandas development team, 2025) dataframe, with each entry containing fields such as course name, course catalogue number, college/graduate, language of instruction, time period(s), is part of, and course description.

2.3 Recommender System

The recommender is memory based as it uses and stores all the vectors directly. Additionally, filtering is content-based. The preprocessing done on the data was, firstly, dropping the rows within the dataframe which had at least one column empty. Secondly, stopwords and punctuation were removed from the dataframe and all columns except the course catalogue number column were changed to lower-case.

The first attempt of making the recommender was with the Doc2Vec vectorizer. Before the vectorization the WortPunktTokenize tokenizer (from the NLTK library (Bird et al., 2009)) was used on each of the columns to tokenize the data. Then all of the relevant columns were combined into one new column of the dataframe. Next, vectorization was performed by using the Doc2Vec vectorizer from gensim library (Řehůřek & Sojka, 2010). The parameters of the Doc2Vec model were: vector_size = 100, window = 5, epochs = 5, min_count = 5, and alpha = 0.05. However, the results were worse than we expected, so for that

reason we changed our approach by selecting a new vectorizer.

We then proceeded to create a base model using the TF-IDF vectorizer from sklearn. There was no tokenization done beforehand as it is done by the vectorizer. Therefore, the 4 columns which were mentioned while talking about the Doc2Vec model were concatenated, forming a new column but without tokenization beforehand. This new column was then passed to the model. This gave us preliminary results that we subsequently built upon by using the Sentence BERT Transformer.

Subsequently, the recommend_courses function takes in the course catalogue numbers as a list and with that finds the corresponding vectors from the TF-IDF/Sentence BERT Transformer vector space. Afterwards, the mean of those retrieved vectors is calculated. Therefore, it was possible to calculate cosine similarity between the courses that the given student has taken before and all the other vectors (courses) which are present in the preprocessed dataset. Before returning the top 5 results the courses which were in the input list are removed from the cosine similarities.

3. EVALUATION OF RESULTS

We used the Normalised Discounted Cumulative Gain (NDCG) metric to evaluate the recommendations, considering the hierarchical ranking of the courses, and the Intra-list Diversity (ILD) metric to analyze the difference between the recommended items (Zangerle & Bauer, 2022). With our recommender, we wanted to strike a balance between precision and diversity. The latter, when lacking, makes the recommendations appear repetitive and uninteresting.

3.4.1 NDCG Metric

We evaluated the model with the NDCG metric, using ndcg_score from the sklearn metrics module (scikit-learn developers, 2025). Our recommender aims to display the courses with decreasing relevance. Therefore, we chose NDCG to account for the hierarchical order in evaluating the accuracy.

The assignment of the course relevance to a specific track was done manually, subjectively incorporating the context of other recommended items - the course might have been marked as more/less relevant depending on what other courses appeared on the recommendation list.

The hierarchical evaluation showed the overall better performance of the SentenceBERT Transformer model over the TF-IDF Vectoriser; however, both models performed well, achieving an accuracy of around 90% only average. The best NDCG scores were obtained for physics, biomedical, law/IR, and film majors. Comparing the track-specific effectiveness, TF-IDF presented better results in art history/art, combining these

Table 1. NDCG metric results

	TF-IDF Vectoriser	Sentence BERT Transformer
Physics	0.991	0.968
Biomedical	0.994	0.982
Law/IR	0.912	1.000
Film	1.000	1.000
Art history/history	0.879	0.796
Cultural analysis	0.715	0.861
Math/Information	0.509	0.595

two more accurately, while SentenceBERT Transformer gave better recommendations for cultural analysis. Both models underperformed outstandingly on recommending the courses for the math/information track, achieving around 60% score.

3.4.2 Diversity Metric

We computed the diversity metric as the mean of cosine distance between the recommended items. We aimed at comparing the diversity of the output courses between the two models, which can be seen in Table 2.

The results showed a significant difference between the TF-IDF Vectoriser and Sentence BERT Transformer with the average diversity scores of around 0.700 and 0.250, respectively. Both models had the lowest diversity for Law/IR courses, which aligns with their high NDCG scores in that track. Although both models have high NDCG scores, the TF-IDF vectoriser receives much higher ILD scores, indicating higher diversity between recommendations. This means that while the Sentence BERT Transformer's recommendations maintain a high cosine similarity between each other, the TF-IDF recommendations have a large difference in cosine similarity, while each entry individually maintains a high cosine similarity with the averaged vector of the course inputs. This is an intrinsic difference caused by the way the two approaches handle the vector space.

Table 2. Diversity metric results

	TF-IDF Vectoriser	Sentence BERT Transformer
Physics	0.894	0.366
Bio/environment	0.901	0.234
Law/IR	0.344*	0.164*
Film	0.686	0.247
Art history/history	0.658	0.242
Cultural analysis	0.786	0.184*
Math/Information	0.882	0.334

* indicates significantly lower diversity compared to other outputs

4. DISCUSSION

While the recommender captures the main dependencies and makes accurate choices, there are several limitations to the project.

The length of the course descriptions appeared to be quite short and irregular concerning their structure. A relatively small amount of meaningful tokens to analyze impairs the process of detecting similarities between the courses and fails to recognize patterns that would indicate academic track. We suspect this limitation may have caused our preliminary results to appear random.

Another limitation concerns the lack of a user network, which would allow us to implement collaborative-based filtering together with our content-based approach. The rationale for implementing both mechanisms flows from research evidence that such a combination is the most effective (Geetha et al., 2018; Widayanti et al., 2023).

Due to time limitations, we could not implement all the characteristics that the recommender system for AUC students would benefit from. Therefore, we list ideas and justifications for their relevance for future research. Primarily, to make the recommender more applicable to the real AUC student experience, we propose extending the dataset to VU courses,

which are part of the possible offer for AUC undergraduates. Importantly, future work should include AUC graduation requirements, taking into account Academic Core courses, Electives, and Other Major Requirements. Additionally, we propose adding extra input for interests, which would extend the recommender to choosing the most accurate courses from other majors.

References

References

Algarni, S., & Sheldon, F. (2023). Systematic Review of Recommendation Systems for Course Selection. *Machine Learning and Knowledge Extraction*, *5*(2), 560–596. https://doi.org/10.3390/make5020033

Geetha, G., Safa, M., Fancy, C., & Saranya, D. (2018). A Hybrid Approach using Collaborative filtering and Content based Filtering for Recommender System. *Journal of Physics: Conference Series*, *1000*(1), 012101. https://doi.org/10.1088/1742-6596/1000/1/012101

Morsomme, R., & Alferez, S. V. (2019). *Content-Based Course Recommender System for Liberal Arts Education.* International Educational Data Mining Society; International Educational Data Mining Society. e-mail: admin@educationaldatamining.org; Web site: http://www.educationaldatamining.org . https://eric.ed.gov/?id=ED599195

Richardson, L. (n.d.). *Beautiful Soup (Version 4.13.3)* [Computer software]. https://www.crummy.com/software/BeautifulSoup/bs4

Roy, D., & Dutta, M. (2022). A systematic review and research perspective on recommender systems. *Journal of Big Data*, *9*(1). https://doi.org/10.1186/s40537-022-00592-5

scikit-learn developers. (2025). *scikit-learn (Version 1.6.1)* [Computer software]. https://scikit-learn.org/

Selenium Contributors. (2025). *Selenium WebDriver for Python (Version 4.33)* [Computer software]. https://www.selenium.dev

The pandas development team. (2025). *pandas (Version 2.2.3)* [Computer software]. https://pandas.pydata.org

Widayanti, R., Chakim, M. H. R., Lukita, C., Rahardja, U., & Lutfiani, N. (2023). Improving Recommender Systems using Hybrid Techniques of Collaborative Filtering and Content-Based Filtering. *Journal of Applied Data Sciences*, *4*(3), 289–302. https://doi.org/10.47738/jads.v4i3.115

Zangerle, E., & Bauer, C. (2022). Evaluating Recommender Systems: Survey and Framework. *ACM Computing Surveys*, *55*(8). https://doi.org/10.1145/3556536

Zhao, Y., Wang, Y., Liu, Y., Cheng, X., Aggarwal, C., & Derr, T. (2023). *Fairness and Diversity in Recommender Systems: A Survey*. ArXiv.org. https://arxiv.org/abs/2307.04644