ual: creative computing institute

IU000135 PERSONALISATION AND MACHINE LEARNING

[1]MINI PROJECT REPORT YIFAN FENG

Project Overview

Inspired by Assignment 2 of designing a university program recommendation system, I explored two different approaches (i.e., content-based and collaborative filtering methods) of constructing recommenders. My initial idea was to provide course recommendations for fresh students to register study units (electives). Due to the limitation of chosen dataset's data structure (see discussion), I narrowed down on two categories of target audience:

1. New students who want to specialise in one academic field and deepen their knowledge through similar knowledge training
2. Registered students look for potential study buddies based on similar study interest

To evaluate two recommendation models, I also conducted a comparative case study of exploring two online MOOCs (Coursea and Udemy) and discussed several enhancement methods in the following sections. Overall, this project investigates course recommendation mechanism for both commercial and academic purposes.

Data & Experiment Design

My project utilizes three open data source from Kaggle website (Batra, 2022; Kapoor, 2022) and Harvard Dataverse (HarvardX, 2014). Figure 01 and 02 demonstrate an overview of pre-processed dataset (e.g., remove irrelevant metadata such as URLs).

```
[11]: df_new_data.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 4505 entries, 0 to 719
Data columns (total 5 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Course Name         4505 non-null   object
 1   University          4505 non-null   object
 2   Course Difficulty   4501 non-null   object
 3   Course Description  4497 non-null   object
 4   Course Objective    4490 non-null   object
dtypes: object(5)
memory usage: 211.2+ KB
```

(Fig 01. Merged MOOCs)

```
[3]: df_harvard_edx.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 338223 entries, 0 to 338222
Data columns (total 20 columns):
 #   Column             Non-Null Count   Dtype
---  ------             --------------   -----
 0   course_id          338223 non-null  object
 1   userid_DI          338223 non-null  object
 2   registered         338223 non-null  int64
 3   viewed             338223 non-null  int64
 4   explored           338223 non-null  int64
 5   certified          338223 non-null  int64
 6   final_cc_cname_DI  338223 non-null  object
 7   LoE_DI             294104 non-null  object
 8   YoB                299719 non-null  float64
 9   gender             305262 non-null  object
 10  grade              312333 non-null  object
 11  start_time_DI      338223 non-null  object
 12  last_event_DI      185992 non-null  object
 13  nevents            178945 non-null  float64
 14  ndays_act          195713 non-null  float64
 15  nplay_video        33277 non-null   float64
 16  nchapters          193758 non-null  float64
 17  nforum_posts       338223 non-null  int64
 18  roles              0 non-null       float64
 19  incomplete_flag    77385 non-null   float64
dtypes: float64(7), int64(5), object(8)
memory usage: 51.6+ MB
```

(Fig 02. HarvardX)

[1]https://git.arts.ac.uk/21036265/Personalization_miniproject

As Kaggle's data source contains more textual information such as course description and objective and less numerical entries such as rating scores, I chose to implement a content-based recommendation approach to answer the question: ***what following courses can student take based on study interest?*** In brief, this recommendation model is to classify courses with specific words and terms that are extracted from course difficulty, description and objective columns and relevance ranking (amongst multiple items) is based on vector space method through word vectorization and similarity calculation (e.g., cosine similarity). Detailed text processing and model structure are documented in the coding script.

The second dataset consists of five courses and metadata (e.g., registration and completion log) of 33k students. With this data structure, I shifted my questions to: ***which students will enroll the same course?*** The second recommender model utilizes an item-item collaborative filtering method to look into target students (who sign up and finish with their class) and compute how similar they are to one course and recommend potential peers enrolled in the selected courses. Similarity is assessed by DataFrame.corrwith( ) function which calculates pairwise correlation. The purpose is to pair up students with similar study interest.

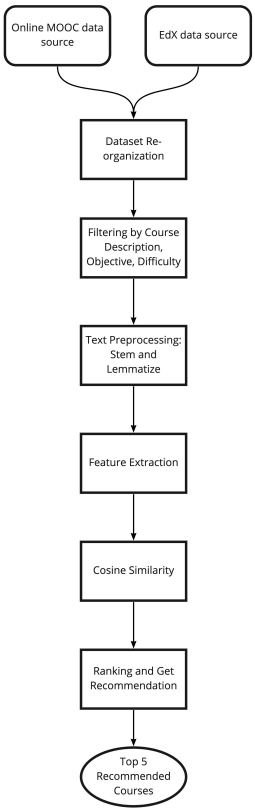


Diagram 01. Content-based Filtering

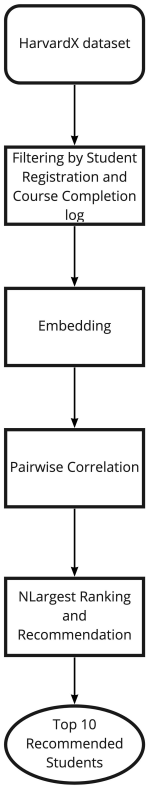


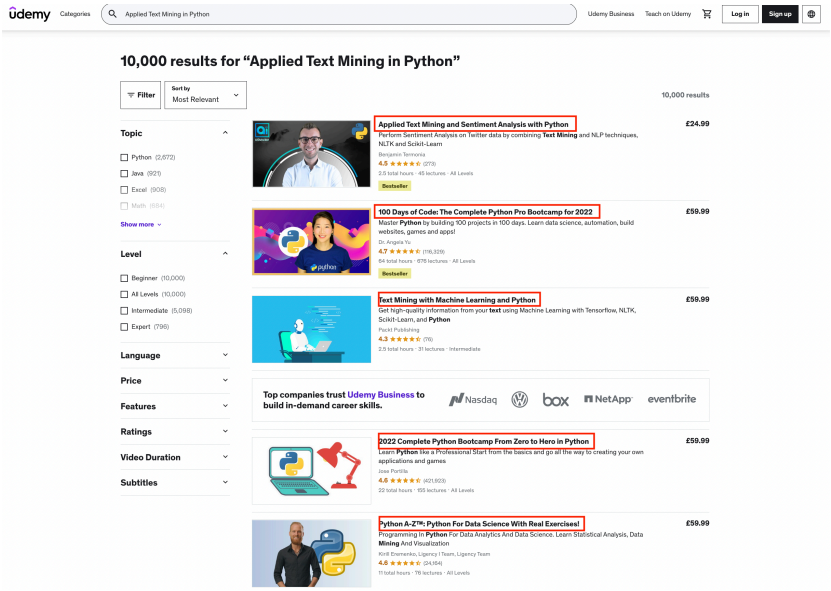
Diagram 02. Collaborative Filtering

Discussion

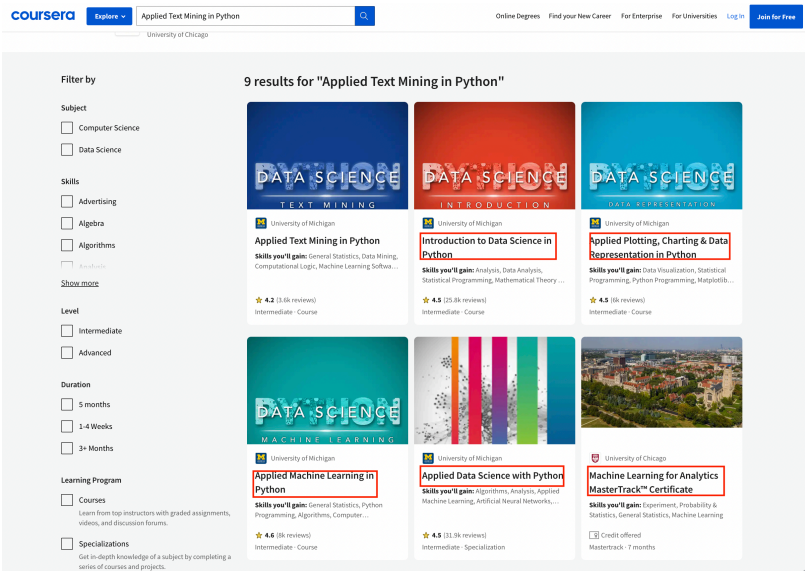
```
recommend_top5("Applied Text Mining in Python")
```

```
Applied Social Network Analysis in Python
Introduction to Data Science in Python
Applied Machine Learning in Python
Applied Plotting, Charting & Data Representation in Python
Clinical Natural Language Processing
```

(Fig 03. Model prediction: top 5 outcome)



(Fig 04. Udemy search result)



(Fig 05. Courera search result)

```
recommend_top5("App Design and Development for iOS")
```

```
Build Your Own iOS App
iOS App Development Basics
Best Practices for iOS User Interface Design
Build a Firebase Android Application
Toward the Future of iOS Development with Swift
```

(Fig 06. Over-specaliased output)

Figure 03 to 05 shows the comparison of the content-based model prediction and online MOOC search on one course query. When inquiring the model about suggestions for “Applied Text Mining in Python” course, four out of five recommendations (top 5) contains the keyword “Python” and the module “Clinical Natural Language Processing” is seen a specialization course under Text Mining domain. Online results demonstrate similar recommendation pattern: the outcome either includes “Python” (see Udemy result) or a new course falls under the same field of enquiry (see Coursera result). In their research, Ma et al. (2017) proposed the semantic similarity approach to recommend courses. They measured similarity between course descriptions and grouped them based on averaged course vectors (similarity matrix). My model is a simplified implementation of this approach: Ma et al.(2017) built a Document-to-Vector (Doc2Vec) matrix after word vectorization to assign mean value of word vectors to each course. Their approach helps to compute similarity value at a large scale (as it sees all given documents such as the entire course description as one entity) and construct a real-time online evaluation system to adjust similarity scores in the database.

Yin et al. (2020) suggested investigating student’s learning behaviors to improve personalized course recommendation. The combination analysis of course content, browsing history, demographic stats and inter-relationship of MOOC courses helps to accommodate both first-time and frequent users. An overall architecture of this method is to (1) access browsing history to identify and cluster user interest; (2) and use demographic data and course-course relationship to adjust probability training for the model prediction. Compared to this approach, my content-based model can only give suggestions based on current interest (assume that user knows their study interest and expectation) and offer low-novelty level (aka overspecialization) outcome (see figure 06).

```
result = get_student_recommendations(df_course_matrix, 'MHxPC130597661')
```

```
display(result)
```

	student_id	correlation
1	MHxPC130000006	1.0
7	MHxPC130000026	1.0
29	MHxPC130000109	1.0
31	MHxPC130000114	1.0
39	MHxPC130000137	1.0
42	MHxPC130000140	1.0
43	MHxPC130000141	1.0
51	MHxPC130000168	1.0
58	MHxPC130000188	1.0
64	MHxPC130000207	1.0

(Fig 07. Model prediction: top 10 students)

course_id	registration_num	view_number
HarvardX/CS50x/2012	169621	106086
HarvardX/ER22x/2013_Spring	57406	32161
HarvardX/PH207x/2012_Fall	41592	24279
HarvardX/PH278x/2013_Spring	39602	15016
HarvardX/CB22x/2013_Spring	30002	16314

(Fig 08. Item-item data exploration)

As aforementioned, due to HavardX's data structure, my collaborative filtering model is used to reveal potential students with similar study interest. Figure 07 demonstrates top ten student matching results (input is student ID number). In this item-item collaborative filtering, one student ID is reckoned as one item and similarity matrix is calculated on course completion log (i.e., "view_number" in figure 08). One of the weaknesses of this approach roots in the imbalanced data entries (32k student ID vs. 5 courses). Another disadvantage is that this approach cannot handle large sparse data (fill zero when data is missing). Together when recommending peers to the student based on course selection, thousands of students will get a full correlation score (corr. = 1) and elevate the difficulty of evaluating relevance between two items (students). That said, with an ideal database, this method is more suitable to suggest courses for users: the algorithm has an advantage of using student's study history/rating to recommend other similar items and thereby match the user expectation and needs. Obeidat et al. (2019) used this idea to create a recommender system: they first computed similarity and dissimilarity of course history between students and clustered students based on grade performance. Their recommendation

One common disadvantage of collaborative filtering system is the "cold-start" issue, that is, when the recommendation model fails to create an embedding for new item-user pairs and thereby cannot process item enquiry. For example, when a new course (without being studied or browsed) is added to the dataset, this course will be less likely sent to the recommendation list. To resolve the issue, Pan et al. (2021) presented a solution of incorporating implicit data entries that include regional distribution of users, login time on weekdays and weekends, user loss rate (last active log) to update course recommendation attribute for users. And for commercial applications, they also recommended that MOOC provider should highlight popular courses (free or paid) to new users and decrease paid course recommendation for loyal customers. Following their logic, in the university context, I suggest that the university's course recommendation system can consider promoting courses with higher averaged marks for students who have a lower GPA. This suggestion is based on the observation that higher education students seek rating forums or website (e.g., RateMyProfessor.com) to review teaching staff's assessment outcome or look into previous year's grade distribution summary before signing up classes. In practice, Adilaksa and Musdholifah (2021) investigated the ways of integrating historical student study results into course recommendation systems. In their content-based model, when calculating cosine similarity between course syllabus, they chose to update similarity weight by weighting student's transcript and assign weighted value (a combination of course syllabus similarity and course score on average) to each course. This supports students who want to elevate their GPA by selecting electives with generous marks.

Similar to the first model's evaluation method, the second model was assessed empirically: I used student ID to locate course completion history in the original dataset and cross-checked with input student's study status. This objective review provides an

insight into the accuracy of model performance. For future work, I can conduct a more subjective approach by reviewing recommendation's *novelty, diversity, serendipity and relevancy* (Adilaksa and Musdholifah, 2021). The strength of subjective assessment helps to identify (1) the level of appropriateness of recommended courses; (2) the level of student expectation/surprise feeling of recommendation output.

Summary

The coding component is less complex due to my knowledge gap on data science and the chosen dataset. Here are some main take-aways from reviewing papers and exploring my own models:

1. Content-based recommendation is more suitable for course recommendation algorithms. This approach has the strength when new courses (no historical review) need to be promoted.
2. For university use, course recommendation should consider student history of study marks when computing weights for course similarity. However, collaborative-filtering based on grade clusters might not be the best method as it limits students who want to achieve a higher score and do not want to be defined by previous academic performance.
3. For commercial platforms, hybrid application might be the best recommendation approach. That said, as hybrid system includes both advantages and disadvantages of content-based and collaborative-filtering methods, a situation of "over parameterization" (e.g., process a large quantity of metadata) might occur.
4. Overall, there is no perfect model for course recommendation. Many subjective factors such as practicality, interest, goal, grade, teacher reputation are involved when determining a course.

Reference

[Dataset]

Batra, A., 2022. *Online MOOC*. [online] Kaggle.com. Available at: <https://www.kaggle.com/datasets/ayushbatra/online-mooc> [Accessed 26 June 2022].

HarvardX. 2014. *HarvardX Person-Course Academic Year 2013 De-Identified dataset, version 3.0*. [online] Available at: <https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/26147> [Accessed 26 June 2022].

Kapoor, K., 2022. *EdX Courses Dataset 2021*. [online] Kaggle.com. Available at: <https://www.kaggle.com/datasets/khusheekapoor/edx-courses-dataset-2021> [Accessed 27 June 2022].

[Academic Paper]

Adilaksa, Y., Musdholifah, A., 2021. Recommendation System for Elective Courses using Content-based Filtering and Weighted Cosine Similarity, in: 2021 4th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI). Presented at the 2021 4th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), pp. 51–55. https://doi.org/10.1109/ISRITI54043.2021.9702788

Ma, H., Wang, X., Hou, J., Lu, Y., 2017. Course recommendation based on semantic similarity analysis, in: 2017 3rd IEEE International Conference on Control Science and Systems

Engineering (ICCSSE). Presented at the 2017 3rd IEEE International Conference on Control Science and Systems Engineering (ICCSSE), pp. 638–641. https://doi.org/10.1109/CCSSE.2017.8088011

Obeidat, R., Duwairi, R., Al-Aiad, A., 2019. A Collaborative Recommendation System for Online Courses Recommendations, in: 2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML). Presented at the 2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML), pp. 49–54. https://doi.org/10.1109/Deep-ML.2019.00018

Pan, Z., Zhao, L., Zhong, X., Xia, Z., 2021. Application of Collaborative Filtering Recommendation Algorithm in Internet Online Courses, in: 2021 6th International Conference on Big Data and Computing. Presented at the ICBDC 2021: 2021 6th International Conference on Big Data and Computing, ACM, Shenzhen China, pp. 142–147. https://doi.org/10.1145/3469968.3469992

Yin, S., Yang, K., Wang, H., 2020. A MOOC Courses Recommendation System Based on Learning Behaviours, in: Proceedings of the ACM Turing Celebration Conference - China. Presented at the ACM TURC'20: ACM Turing Celebration Conference - China, ACM, Hefei China, pp. 133–137. https://doi.org/10.1145/3393527.3393550