General Information

Names and student numbers

Martin Michaux (5701333),
Alexandru Dumitru (5074711),
Armin Korkić (4713052),
Giovanni Trinchillo (5819873)

Introductory description of the topic

Our Multimedia Search and Recommendation project proposal aims to address the challenges faced by Recommender Systems (RS) in the context of course enrolment [13]. We propose a novel solution that leverages *bipartite graph structures* [17], comprising user and course graphs, to provide personalized recommendations. By incorporating user reviews and previous records, our system aims to overcome common problems encountered in matrix-based recommender systems.

In traditional RS, *sparsity* arises due to the limited interactions between users and the vast number of available courses. However, our graph-based approach mitigates this issue by representing only existing connections, eliminating the need to consider missing ones [6]. Additionally, we tackle the *cold start* problem, where insufficient data exists for new users or items, by utilizing similarity measures between users or items, even in the absence of direct interaction data [30].

The utilization of bipartite graphs provides several advantages, including *scalability* for large-scale systems, as it eliminates the need to compute recommendations for all possible user-item pairs [7]. By capitalizing on the inherent structure and relationships within the graph, our system is designed to generate personalized and accurate course recommendations, enhancing user satisfaction and engagement.

Through our project, we aim to demonstrate the effectiveness of our graph-based RS solution, leveraging the latest research and techniques in the field. By combining user reviews, previous records, and graph-based approaches, we aspire to revolutionize the course recommendation landscape, addressing common challenges and improving the overall learning experience for students.

Motivation

The exploration of a personalized Recommender System (RS) for course enrollment is a critical subject with profound implications for both students and academic institutions. [20] The primary objective of this investigation is to adopt an innovative approach to RS that addresses some of the challenges it faces, particularly in the context of course recommendations. This domain presents a heightened sense of these challenges due to the annual influx of new students, who represent new users in the system. These new users require a great number of recommendations but have no data to their name, intensifying the impact of the cold-start problem. Moreover, students tend to interact with only a limited subset of the available courses, resulting in sparse data.

An additional motivation stems from the domain-specific nature of Recommender Systems (RS), as it is often challenging or infeasible to directly transfer ideas and or methodologies from one domain to another. In the education sector learners are still being provided with static and predefined patterns of learning courses, tasks, materials, and objects in spite of the fact that learners differ in characteristics such as learning interests, objectives, needs, skills and personalities. [9]

The last factor that motivated us to choose specifically course enrolment is to help institutions improve the quality of their academic programs. By analysing student reviews of courses, institutions can gain valuable insights into which courses are most popular and effective and which areas need improvement. This can help institutions to tailor their course offerings to better meet the needs and interests of their students.

The urgency of addressing this issue is high, particularly given the challenges posed by the COVID-19 pandemic. With the increasing need for online learning and more efficient systems for course enrolment, the development of a personalized recommender system can help improve the overall quality of education for students and institutions.

Link to CS4065

The primary objective is to address common challenges encountered in RS and specifically within the domain of course recommendation, while also enhancing the effectiveness of user recommendations. Traditional methods such as collaborative filtering or content filtering solely rely on user or content information, respectively [31]. In contrast, this project focuses on a recommendation method that integrates both data sources into a graph structure. The connections within the graph are established based on the course reviews generated by users and the content similarity between courses, enabling a more comprehensive and robust representation of the data.

One prominent challenge encountered in RS is the presence of data sparsity within datasets, which refers to a scenario where a significant portion of the matrices comprises empty cells [28]. This sparsity issue is particularly apparent in course enrolment datasets, as each student typically engages in only a limited subset of the total available courses. To address this challenge, the utilization of graphs proves advantageous. Unlike matrices, graphs do not explicitly represent the absence of connections, eliminating the need to allocate cells that contain zero values for non-existent links. By leveraging graphs, the issue of data sparsity is mitigated, allowing for a more efficient and concise representation of the dataset.

This project exhibits characteristics of a hybrid RS, as it employs two distinct graphs to generate personalized recommendations. One graph comprises connections between users who exhibit similar review patterns for courses, while the other graph includes links between courses that possess similar attributes. The goal is by utilizing this hybrid approach, the recommendation system achieves broader coverage of potential recommendations and has the potential to enhance the accuracy of its recommendations.

Bipartite Graph-based Recommender System for Addressing Cold Start and Sparsity

Martin Michaux (5701333),
Alexandru Dumitru (5074711),
Armin Korkić (4713052),
Giovanni Trinchillo (5819873)

Proposal summary

This research project presents a groundbreaking approach to course recommendations by developing a novel bipartite graph-based recommender system (RS) [34, 17] that utilizes both user previous records and course reviews. Our innovative methodology aims to address the cold start problem and sparsity issues [6, 30, 7] prevalent in recommendation algorithms, ensuring accurate and personalized course-user recommendations.
Unlike traditional collaborative filtering methods [13, 10] that rely solely on matrix/tabular data, our approach incorporates a bipartite graph structure. This graph consists of interconnected nodes representing users and courses, enabling us to capture the intricate relationships between them. By leveraging both user and course reviews, we can overcome the limitations imposed by data sparsity and the challenges of cold start scenarios.
Integrating the renowned PageRank [23] algorithm within the bipartite graph framework further enhances the novelty of our research. Originally devised for web search ranking, PageRank proves adaptable for recommender systems in this context. By treating the bipartite graph as a web-like network, we assign importance scores to users and courses, enabling us to generate more accurate and reliable recommendations.
The innovative nature of our approach lies in its ability to combine user reviews and course information to generate personalized recommendations. This unique integration allows us to tap into a wealth of knowledge present in both user experiences and course characteristics, ultimately enhancing the quality of recommendations.
Our research proposal not only holds significant implications for industries, particularly online education platforms, where accurate course recommendations are vital, as they facilitate personalized learning experiences, but also contributes to the academic community by addressing the challenges of cold start problems and sparsity in Recommender Systems [10]. By tailoring personalized recommendations, our system enhances user satisfaction and engagement while providing businesses with improved user retention and conversion rates.
To evaluate the effectiveness of our novel RS, we will conduct comprehensive experiments using a real-world dataset sourced from Coursera, a prominent online platform offering a wide range of courses. This dataset represents courses that closely resemble real-world offerings, ensuring the relevance and applicability of our research. By comparing the performance of our bipartite graph-based approach with traditional collaborative filtering techniques, we aim to demonstrate the superior accuracy and utility of our recommendations. Evaluation metrics such as precision, recall, hitrate and mean average precision, widely used in the field of recommender systems [33], will be employed to quantitatively assess the performance of our RS.

Link to video and repository

Gitlab repository:
https://gitlab.ewi.tudelft.nl/cs4065/2022-2023/team-22/msr_project_group22

Youtube video:
https://youtu.be/BhjHJZzbI6k

Extended synopsis of the project proposal

1. Problem description

In the realm of education, the need for accurate and personalized course recommendations is critical. Traditional collaborative filtering methods [10], which rely solely on matrix/tabular data, face two significant challenges: the cold start problem and data sparsity [6, 30, 7]. These obstacles severely limit the effectiveness of recommendation algorithms in generating meaningful and relevant course suggestions for users.

The cold start problem arises when new users join the platform or when new courses are introduced, leading to a scarcity of historical data on which recommendations can be based. Similarly, data sparsity occurs when there is a lack of user interactions or course reviews, making it difficult to establish reliable connections and infer preferences.

Addressing these challenges is essential to enhance user experiences, increase engagement, and improve the overall success of online education platforms. Thus, there is a pressing need for a novel recommender system (RS) solution that can effectively overcome the cold start problem and data sparsity while delivering accurate and personalized course-user recommendations.

The proposed solution of a groundbreaking bipartite graph-based RS [34, 17] that incorporates user previous records and course reviews holds significant relevance and significance in this context. The innovative integration of the renowned PageRank [23] algorithm within the bipartite graph framework further enhances the solution's effectiveness. By assigning importance scores to users and courses based on their interactions within the graph, the RS can generate accurate and reliable recommendations, addressing the limitations of traditional methods. By addressing the cold start problem and data sparsity [10], the RS will provide users with tailored recommendations, improving user satisfaction, engagement, and learning outcomes.

2. Previous work

Recommender systems dedicated to course recommendations have been getting more attention in recent years; in this section, we will provide an overview of the current methods and their challenges.

Content-Based Recommender System (CBRS): This type of recommender systems generally compares user preferences against the contents of the items present in the system. [10] These approaches focus on analysing the content of courses – such as course descriptions, learning objectives, or syllabi – and then leverage techniques such as natural language processing (NLP) and information retrieval to extract relevant features.

Since these systems influence future recommendations by means of the user's own ratings, a pure content-based approach has several fundamental drawbacks : limited content analysis, difficulty to apply non-textual data, risk of overspecialization by recommending only items that bear strong resemblance to those the user already knows and sparsity of data [1] [25]. For these reasons, most of the CBRS proposed usually take a hybris approach [21].

Collaborative Filtering-Based Recommender System (CFBRS): Collaborative filtering technique works by building a database (user-item matrix) of preferences for items by users. It then matches users with relevant interest and preferences by calculating similarities between their profiles to make recommendations. CFBRS can produce either predictions or recommendations: a prediction is a numerical value expressing the predicted score of specific item for a certain user, while recommendation is a list of top N items that the user will like the most [13]. CFBSR have several drawbacks, such as cold-start, sparsity and gray sheep problems. Gray sheep refers to poor predictions due to the lack of similar users with similar preferences in the system [8]. A solution proposed in the same publication is to use K-means clustering algorithm offline, with different centroid selections and different distance measures.

Despite these problems, it is one of the more mature and widely used approaches, with a large body of literature available [14]. An example of user-based CFBRS is CourseAgent [8], which is a community-based course RS that relies on user participation and aims to motivate them, while [16] proposes a model-based one.

Knowledge-Based Recommender System (KBRS): The knowledge-based approaches use domain knowledge to generate recommendations which meet user preferences [32]. The main drawbacks in KBRS are difficulty of data acquisition and representing the domain knowledge in a machine-readable structure, and it needs extensive efforts to extract the knowledge.

Hybrid Techniques: Hybrid RS are a combination of different recommendation techniques, with the aim of overcoming or reducing the impact of the drawbacks of each single technique and improve the quality of the final recommendation. One approach is to combine CFRS and CBRS [13] to achieve better performance. Different ways to create hybrid approaches by combining CF and content-based filtering have been proposed [19]. Another approach combined the previous hybrid one with Ontologies to improve the final course recommendation to users [9]. The main drawbacks of these approaches are in integrating the different techniques and in dealing with the increasing complexity that the use of multiple techniques brings.

3. Scientific contribution

Proposed idea:

In order to address the challenges associated with course RS and general RS, we propose a novel approach utilizing a multi-layer graph. This approach integrates two distinct graphs, namely a user graph and a course graph, to create a comprehensive graph that utilizes the interconnectedness of users, courses, and course interactions. By doing so, our model aims to generate precise and personalized course recommendations. The methodology employed in this project draws inspiration from previous studies conducted in various domains [12]. The primary objective is to tackle specific issues pertaining to RS in general, with a particular emphasis on course recommendation systems.

Innovation:

The most important novelty of this concept lies in the application of established techniques and methodologies to a research domain that currently lacks related studies [12] [24]. We argue that the proposed model shows great potential within the realm of course recommendation for several reasons. For starters, it effectively addresses the cold-start problem inherent in course recommendation systems by leveraging user histories and course similarities to provide recommendations even for users with minimal review data [4]. Secondly, course datasets generally show case high levels of sparsity, with numerous courses available but users participating in only a limited selection [29]. Previous studies have employed graph data structures to mitigate the issues arising from data sparsity [35]. Consequently, the objective of this study is to assess the efficacy of the proposed method in mitigating data sparsity and resolving the cold-start problem within course datasets.

Level of challenge:

Realizing the proposed contribution involves challenges such as applying the main topic segmentation to course descriptions, sentiment analysis on user reviews, and managing the computational complexity of calculating similarities between users. To efficiently accomplish these tasks, advanced techniques in natural language processing, memory management, and computational approaches are required. These measures are important to ensure precise sentiment analysis and to handle the substantial dataset comprising course-user reviews. Moreover, acquiring a dataset derived from real-life course data would significantly enhance the quality and validity of our study.

Level of impact:

The proposed research aims to address the challenges encountered in the field of course recommendations, as well as certain general issues within recommendation systems [31]. As previously discussed, we argue that this model has the potential to mitigate the cold-start and sparsity problems present in the domain of course recommendations. This is achieved by integrating user-review data with course-similarity information, thereby generating a broader range of potential options for new users, consequently alleviating the cold-start problem. Additionally, by utilizing a graph-based data structure to store and manage the relevant information, we can effectively handle the data sparsity inherent in the dataset, as our calculations consider only the edges that are present in the system.

4. Experimental setup

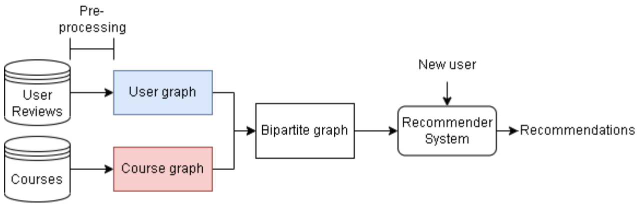


Figure 1: Recommender System Pipeline

4.1. Data processing

The datasets utilized in this research were sourced from Kaggle[1] and encompassed data related to Coursera[2], an online learning platform. One dataset contains historical user interactions with various courses, while the other dataset provides detailed information about these courses. The first one includes ratings, dates, and review texts associated with each user review and, conversely, the latter encompassed attributes such as the course name, institution, level, overall rating, description, and set of skills. To ensure computational efficiency, considering the computational expense of sentiment analysis, we opted to select a subset of 1000 users from the combined datasets consisting of 287,808 distinct users and 1,454,711 reviews. This reduction in the number of users while maintaining a representative sample of 149,908 reviews was necessary due to the complexity associated with sentiment analysis. However, thanks to the scalability advantage of the graph-based structure, the impact of reducing the number of users was mitigated, making it a viable approach for streamlined processing and analysis.

4.1.1 User Graph

We incorporated sentiment analysis using the Twitter Roberta model[18], trained on 58M tweets and fine-tuned with the Tweet Eval benchmark[3], to capture subjective opinions and attitudes expressed by users beyond numerical ratings[26]. This addresses the cold start problem by leveraging qualitative information from early reviewers and enhances personalization, resulting in more accurate and relevant recommendations, leading to improved user satisfaction and engagement. To establish connections between similar users in the user-user graph, we employed a similarity matrix approach[2]. The thresholds were determined based on achieving an optimal sparsity level in the graph, striking a balance between meaningful user similarities and graph sparsity. By computing the similarity matrices using the tuned thresholds, we identified pairs of users with sufficient similarity. Links were created between user pairs when the number of thresholds surpassed a specified overall

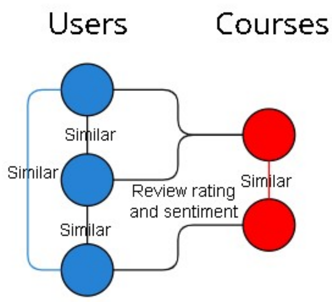
[1]https://www.kaggle.com/datasets/imuhammad/course-reviews-on-coursera
[2]https://www.coursera.org/

threshold. This approach facilitated accurate and personalized recommendations by connecting users with significant similarity across multiple features in the user-user graph.

4.1.2 Course Graph

The course-course graph construction followed a similar procedure to the user-user graph, enabling their integration into the bipartite graph. To incorporate course descriptions, we employed Latent Dirichlet Allocation (LDA)[15] to model the main topics within the descriptions. This inclusion enriches the graph by capturing key themes and content covered in each course. We generated a similarity matrix for the course features, including institution, level, rating, LDA-derived topics, and skills. By applying specific thresholds for each feature and an overall threshold, we established links between courses in the graph. This approach has the same benefits as stated in the section 4.1.1.

4.2. Bipartite Graph



To create the bipartite graph incorporating the user-user and course-course graphs, we adopted a straightforward approach inspired by the work [22] which investigates how to build a user-item bipartite graph. We established connections between users in the user graph and the courses in the course graph that the users had taken. Each edge in the bipartite graph represented a user-course relationship and was assigned a weight corresponding to the user's rating and sentiment scores for the respective course. This methodology enabled the seamless integration of user preferences and course characteristics, facilitating personalized recommendations within the bipartite graph framework.

4.3. Recommender System

In order to utilize the proposed data structure, a graph-based algorithm known as Personalized PageRank (PPR) was adopted for generating recommendation, idea previously seen in [11] and [23].

PPR, an extension of the original PageRank algorithm used previously by Google [5], introduces an element of personalization. The traditional PageRank computes a global importance score for each node in the graph, but PPR calculates a personalized ranking for nodes in the graph relative to a given source node. This is achieved by introducing a 'personalization vector', which is a probability distribution over all nodes in the graph. The algorithm then performs a random walk, where it probabilistically decides at each step whether to follow an outgoing edge from the current node or to jump back to a node from the personalization vector.

The recommendation pipeline consists of mainly 2 steps: calculation of PPR scores and generation of recommendations. Firstly, a PPR is computed for each user node in the bipartite graph, taking the user node as the 'source node'. Each user node is assigned a personalization vector in which only the user's own node is given a non-zero value. This focuses the attention of the random walk on the immediate surroundings of the source node, in this case, the user. The PPR score is then calculated, with an alpha parameter controlling the probability of continuing along the graph's edges versus teleporting back to a node from the personalization vector(in this case the alpha parameter is set to 0.85). The edge weights, representing the strength of interactions between users and courses, significantly influence this random walk, ultimately affecting the computed PPR scores. The sentiment and rating reviews between users and courses were aggregated into a singular value, in order to be able to apply the PPR. The formula used to aggregate them was:

$$w_{ij} = r_{ij} \cdot positive_{ij} - r_{ij} \cdot negative_{ij}$$

where

- w_{ij} is the weight of the edge connecting user i and course j in the bipartite graph. This weight represents the strength of interaction between user i and course j.
- r_{ij} represents the rating given by user i to course j.
- $positive_{ij}$ and $negative_{ij}$ denote the positive and negative sentiments expressed by user i towards course j, respectively. These sentiments are inferred from user's past reviews.

Subsequently, recommendations are generated based on PPR scores, the weight of user-course interactions, and both user-user and course-course similarities. For user-user similarities, the algorithm adjusts recommendation scores based on the interaction weight between similar users and the course. For course-course similarities, the algorithm boosts the recommendation score for similar courses based on their PPR scores, similarity strength, and user sentiment towards the original course. This approach allows us to incorporate all information from the bipartite graph into a recommender system.

5. Evaluation methodology

The primary objective is to observe whether or not our proposed RS is able to deal with the cold start problem. Therefore, the main experiment conducted will simulate this scenario, of a user without any past interactions.

5.1. Setup

We created a bipartite graph with 527 course nodes, 24874 course edges, 1000 user nodes, 323055 user edges, and 24874 interconnecting edges. For simulating a cold start scenario, we split user nodes into training and test sets. We treated 20% of the users as new by deleting their past interactions and similarities, forming our test set.

5.2. Models for Cold-Start Scenario

As collaborative filtering requires past interactions to generate recommendations, in the cold-start scenario, we use the approach of recommending the most popular course as our baseline. We identify the most popular courses by looking for course nodes with the highest user degree.

Furhtermore, we can't use the PPR model directly as it needs user-specific past interaction data to create a personalization vector. Therefore, we apply a Global PageRank model for these new users. The Global PageRank model computes a global importance score for each node, useful for ranking nodes in the absence of personalized information.

5.3. Experiment

To evaluate the effectiveness of our Global PageRank model and the baseline system, we use a top-N recommendation task, where N is set to 10. We choose the hit rate as the evaluation metric, as it is especially relevant for the cold start scenario. The hit rate measures the proportion of new users for whom our system can correctly predict an interaction, a 'hit' being achieved if a course the user interacted with appears in our top 10 recommendations. Thus, a high hit rate indicates our system's effectiveness in providing relevant recommendations to new users, despite the lack of personalized data. This is crucial as the primary aim of our system is to cater to new users effectively. The experiment was conducted for 100 iterations, with a different random train-test split for each iteration. The hit rates for each method were averaged over these iterations, to ensure our results are not specific to any particular split and can generalize to different user-course interaction scenarios.

Our Global PageRank model surpassed the baseline model in the cold start problem, achieving an average hit rate of 0.488, compared to the baseline model's average hit rate of 0.464.

5.4. General Performance: PPR vs Collaborative Filtering

In the absence of a cold-start situation, we compared the effectiveness of our PPR model with the traditional User-User Collaborative Filtering (CF). For this comparison, we split the users' interactions into training and test sets, assigning 20% of each user's interactions to the test set. This allowed us to evaluate the performance of the models under regular operating circumstances.

Using metrics like Hit Rate, Precision, Recall, F1-Score, and NDCG, we evaluated both models. The PPR model achieved a Hit Rate of 0.408, Mean Precision of 0.235, Mean Recall of 0.604, Mean F1-Score of 0.318, and Mean NDCG of 0.612. The CF model, on the other hand, achieved a Hit Rate of 0.496, Mean Precision of 0.260, Mean Recall of 0.674, Mean F1-Score of 0.358, and Mean NDCG of 0.689. These results indicate that the CF model performed better in general, outside of a cold start scenario.

6. Project objectives, organization and planning

Project Purpose: The purpose of the project is to answer the following research questions; "Can the use of Bipartite Graph in a Course Recommender System address the cold-start and data sparsity problems?". The project will attempt to answer the question by implementing the proposed the model, a baseline model and then running experiments to compare performances. This project is proposed to further extend the research done on RS used for course-data sets, to examine the advantages of using a bipartite graph structure and tackling general recommendation system problems.

Project Resources: This project is planned for 1 academic year, which is roughly 10 months. Therefore, the planning and the resources are divided over a 10-month period. The team will consist of 4 Master students, who will spend 1 day a week towards this project. Other resources which are beneficial towards the project are: separate project rooms, a physical room in which the team comes together to work on the project and the use of external GPU's on separate servers to run the model and experiments on.

Planning:

- 1st Month: Additional Literature Review

  This first month will be spend on concluding the literature review. The team will improve on the work done for this proposal, analyzing additional literature and research regarding the topic of bi-partite graphs and course recommendation systems. The goal of this literature review is to go more in depth on how others researchers have evaluated their models, what methods and models have worked for them and or pitfalls that they might have encountered during their research. One specific need will be to find a method that allows us to evaluate the data sparsity.

- 2nd Month: Data-preprocessing and planning

  The team will choose a data-set for their experiments. But before this data-set can be used it needs to be preprocced. The data needs to be set in the right format and flaws within the data needs to be corrected. Such corrections might entail, removing duplicates, correcting for wrong data types or removing empty rows. Also in this phase will start with the planning of the structure of the code-base, formalizing all the implementational details, like the choice of programming languages.

- 3rd and 4th Month: Implementing the model

During this period the model proper will be implemented, of course some of the planning is expected to bleed into this period, as some of the finer details will have to be reviewed. Besides the project model a baseline model will need to be implemented to be compared to the proposed model.

- 5th, 6th and 7th Month: Running the experiments

  After the model is implemented, experiments needs to be set-up. The code needs to be prepared to be run on the given hardware, these can be mobile workstations or on external servers with dedicated GPUs.The longest time-slot has been allotted to this phase to account for problems that can arise both during the implementation and during the experiments. Also updates or improvement can be done to the model based on preliminary results obtained from the first experiments.

- 8th Month: Evaluating results

  After all the planned experiments have been run, a complete analysis of the data obtained will be done. If any mistakes occured during the experiments, they can be rerun here if given enough time. If everything went well the results can be converted into adequate visual representations.

- 9th Month: Writing the Report

  In the last phase all the findings, the process and the results are formalized in a report, which will follow the classic structure of Introduction, Related Works, Method, Experiments, Evaluation, Results and Conclusion/Discussion.

- 10th Month: Risk Management

  Since likely there will be some issues during one of the previous phases, the plan has been with an extra month in mind, to allow time to fix any possible problems, or act as a buffer in case one of the phases takes longer than planned. If by the 8th month the project is still on schedule, the extra time can be used to plan improvements to the model based on the results of the experiments.

7. Risks and mitigation plan

In conducting research, it is likely that not everything will go according to the plan. It is dificult to foresee all problems that might be encountered, however steps can be taken to prevent certain problems from arising, or make them easier to deal with. This section will discuss a few of these problems and how they can be dealt with.

For starters, it is very likely that the model and the code-base will contain bugs. This is a problem that all computer scientists are familiar with and is inescapable for all software projects. When this occurs there is no ready solution to be used, since this will depend on the context of the bug. However, by following strict coding rules and/or heuristics, they can be prevented or solved more easily. This means following all the good rules of conduct when writing code, such as: keeping the functions as small as possible, writing human-friendly readable code, using proper function and variable names, and having a well-planned and well-documented project structure.

Another type of problems that might occur are those related to data; this type of problems are diverse and it is hard to predict which one will occur; however some are more likely to occur given the experience working on similar data [27]. In no particular order these are some of the most likely: rows may be empty, the data contains data-types which are inconsistent with the rest, some parts of the data are duplicated within the data-set. The plan is to fix these problems in the data-processing part, but fully preventing them will not be easy.

Last but not least, depending on the hardware available there might be problems of scalability. If the data is too large and the model too complex, the calculations might take too long or become unfeabile. If this happens, the dataset will have to be reduced in size, taking care of avoiding an inbalance in the data.

Given the possibility of some of these issues presenting themselves, a month has been kept free to deal with them and still have time for the project to be on schedule

8. References

References

[1] Marko Balabanović and Yoav Shoham. Fab: Content-based, collaborative recommendation. *Commun. ACM*, 40(3):66–72, mar 1997.

[2] Jothi Balasundaram and Pushpalatha Marudappa. Recommendation system—item user matrix: A graph base system. *Journal of Computational and Theoretical Nanoscience*, 15:3044–3048, 09 2018.

[3] Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650, Online, November 2020. Association for Computational Linguistics.

[4] Cesare Bernardis, Maurizio Ferrari Dacrema, and Paolo Cremonesi. A novel graph-based model for hybrid recommendations in cold-start scenarios, 2018.

[5] Monica Bianchini, Marco Gori, and Franco Scarselli. Inside pagerank. *ACM Transactions on Internet Technology (TOIT)*, 5(1):92–128, 2005.

[6] Janneth Chicaiza and Priscila Valdiviezo-Diaz. A comprehensive survey of knowledge graph-based recommender systems: Technologies, development, and contributions. *Information*, 12(6), 2021.

[7] Yue Deng. Recommender systems based on graph embedding techniques: A review. *IEEE Access*, 10:51587–51633, 2022.

[8] Rosta Farzan and Peter Brusilovsky. Encouraging user participation in a course recommender system: An impact on user behavior. *Computers in Human Behavior*, 27(1):276–284, 2011. Current Research Topics in Cognitive Load Theory.

[9] Zameer Gulzar, A. Anny Leema, and Gerard Deepak. Pcrs: Personalized course recommender system based on hybrid approach. *Procedia Computer Science*, 125:518–524, 2018. The 6th International Conference on Smart Computing and Communications.

[10] Deepani B. Guruge, Rajan Kadel, and Sharly J. Halder. The state of the art in methodologies of course recommender systems—a review of recent research. *Data*, 6(2), 2021.

[11] Xiangnan He, Ming Gao, Min-Yen Kan, and Dingxian Wang. Birank: Towards ranking on bipartite graphs. *IEEE Transactions on Knowledge and Data Engineering*, 29(1):57–71, 2016.

[12] Zan Huang, Wingyan Chung, Thian-Huat Ong, and Hsinchun Chen. A graph-based recommender system for digital library. 7 2002.

[13] F.O. Isinkaye, Y.O. Folajimi, and B.A. Ojokoh. Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal*, 16(3):261–273, 2015.

[14] Mahdi Jalili, Sajad Ahmadian, Maliheh Izadi, Parham Moradi, and Mostafa Salehi. Evaluating collaborative filtering recommender algorithms: A survey. *IEEE Access*, 6:74003–74024, 2018.

[15] Hamed Jelodar, Yongli Wang, Chi Yuan, and Xia Feng. Latent dirichlet allocation (LDA) and topic modeling: models, applications, a survey. *CoRR*, abs/1711.04305, 2017.

[16] Mohammad S. Laghari. ”automated course advising system”. 4(1):47–51, 2014.

[17] Xin Li and Hsinchun Chen. Recommendation as link prediction in bipartite graphs: A graph kernel-based machine learning approach. *Decision Support Systems*, 54(2):880–890, 2013.

[18] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

[19] Kasra Madadipouya and Sivananthan Chelliah. A literature review on recommender systems algorithms, techniques and evaluations. *BRAIN. Broad Research in Artificial Intelligence and Neuroscience*, 8(2):109–124, 2017.

[20] Bhaskar Mondal, Om Patra, Sanket Mishra, and Priyadarshan Patra. A course recommendation system based on grades. In *2020 International Conference on Computer Science, Engineering and Applications (ICCSEA)*, pages 1–5, 2020.

[21] Ammar A. Neamah and Amer S. El-Ameer. Design and evaluation of a course recommender system using content-based approach. In *2018 International Conference on Advanced Science and Engineering (ICOASE)*, pages 1–6, 2018.

[22] Alessandro Negro, Amy Hodler, and Mark Needham. *Graph-Powered Machine Learning*. O'Reilly Media, 1st edition edition, 2021. Section 5.2: Creating the bipartite graph for the User-Item dataset.

[23] Phuong Nguyen, Paolo Tomeo, Tommaso Di Noia, and Eugenio Di Sciascio. An evaluation of simrank and personalized pagerank to build a recommender system for the web of data. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, page 1477–1482, New York, NY, USA, 2015. Association for Computing Machinery.

[24] Michael O'Mahony and Barry Smyth. A recommender system for on-line course enrolment: an initial study. 10 2007.

[25] Michael J. Pazzani and Daniel Billsus. *Content-Based Recommendation Systems*, pages 325–341. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[26] Xiaomin Pu, Guangxi Yan, Chengqing Yu, Xiwei Mi, and Chengming Yu. Sentiment analysis of online course evaluation based on a new ensemble deep learning mode: Evidence from chinese. *Applied Sciences*, 11(23), 2021.

[27] Erhard Rahm, Hong Hai Do, et al. Data cleaning: Problems and current approaches. *IEEE Data Eng. Bull.*, 23(4):3–13, 2000.

[28] Francesco Ricci, Lior Rokach, and Bracha Shapira. *Recommender Systems: Introduction and Challenges*, pages 1–34. Springer US, Boston, MA, 2015.

[29] Mirka Saarela and Tommi Kärkkäinen. Analysing Student Performance using Sparse Data of Core Bachelor Courses. volume 7, pages 3–32, 2 2015.

[30] Andrew Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. Methods and metrics for cold-start recommendations. 8 2002.

[31] Lalita Sharma and Anju Gera. A Survey of Recommendation System: Research Challenges. *International Journal of Engineering Trends and Technology (IJETT)*, 1 2013.

[32] John K. Tarus, Zhendong Niu, and Ghulam Mustafa. Knowledge-based recommendation: a review of ontology-based recommender systems for e-learning. *Artificial Intelligence Review*, 50(1):21–48, 1 2017.

[33] Saúl Vargas and Pablo Castells. Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, page 109–116, New York, NY, USA, 2011. Association for Computing Machinery.

[34] Shoujin Wang, Liang Hu, Yan Wang, Xiangnan He, Quan Z. Sheng, Mehmet A. Orgun, Longbing Cao, Francesco Ricci, and Philip S. Yu. Graph learning based recommender systems: A review. *CoRR*, abs/2105.06339, 2021.

[35] Kaige Yang and Laura Toni. Graph-based recommendation system. In *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 798–802, 2018.

Appendix A: Process

<u>*Together:*</u> Explain the process of working towards this final report and product.

For the final report, the team got together for a meeting were we planned the last steps of the project. We took stock of what was done and what needed to be completed. We splitted the work on the basis of what had been done in the midterm, in such a way that nobody was stuck doing the same thing twice. We divided the work in such a way that that everyone worked on the parts where they had a better understanding of. After that, the team kept meeting at least every two or three days to keep track of the progress of each individual member. Concerning the writing of the final report, while each section was assigned to one or two specific members, everyone was asked to review the whole thing at least twice, and give their opinions on possible improvements.

<u>*Per team meber:*</u> Make clear what the contribution and effort was per team member.

<u>Martin:</u> Throughout the project, I took personal responsibility for implementing the course graph and the bipartite graph. In addition to that, I conducted a thorough review of academic papers on general problems and advancements in Recommender Systems. I also collaborated with Armin and Alex in the implementation of the user graph. Furthermore, I made substantial contributions to the writing of the proposal report, specifically in the Proposal Summary, Problem Description, and Experimental Setup sections, while providing overall assistance in the report's development.

<u>*Giovanni:*</u> During the course I had the following responsibilities: With the rest of the team I helped plan the project, starting from choosing the argument to writing the intake, all of which was a team effort. After that I focused on the analysis of the related works regarding the specific domain of recommender systems for courses. During that time period I also worked on the user interface for the prototype of the recommender system, although it was ultimately scrapped when we shifted focus away from the prototype.

In the writing of the final report I took care of updating the general informations provided in the intake, rewriting the parts that needed it, then I wrote the **Previous Work** section and co-wrote the sections **Project objectives, organization and planning** and **Risk Management** with *Armin Korkić*

<u>*Armin:*</u> In this project I had several responsibilities. In the first few week I was responsible for the implementation of the user-user graph. In which I had to write the code-base that processed the online-course data into a graph in which nodes are users and edges indicate the similarity between users. Then later on in the project Martin and Alex also collaborated on the user-user graph.

In this project I also did much literature review, in which I tried to find as much papers and articles that are relevant to this project. And I contributed to the writing of all the reports. In the final reports my biggest contributions are the scientific contribution, the planning and risk mitigation section.

<u>*Alex:*</u> I worked mainly on the implementation of the graph and the recommender system. During the first weeks I helped Martin and Armin with implementing the user-user graph. I performed the sentimental analysis on reviews using RoBERTa pretrained model, and added the weights on the edges between users. Afterwards I looked into literature to see how we could use our bipartite graph to generate recommendation, thus finding papers sugesting the use of PageRank. Therefore, I was mainly responsible for the implementation of the recommender, and the experiments conducted.

Appendix B: Reflection

Martin Throughout this project, I have acquired significant scientific knowledge and technical expertise that have enhanced my understanding of the research domain. From a scientific standpoint, I have expanded my understanding of Recommender Systems, particularly within the realm of course enrollment. Through comprehensive literature review encompassing general issues and recent advancements in Recommender Systems, I have gained valuable insights into the critical challenges confronted in this field and the inventive approaches proposed by fellow researchers.

On a technical level, I have gained hands-on experience in formulating problem statements and identifying the specific challenges associated with course recommendations. Implementing the bipartite graph has provided me with practical skills in working with graph-based structures for recommendation systems. This experience has broadened my understanding of graph theory and its application in the field of recommendation.

Conducting a thorough literature review has been an integral part of this project, enabling me to explore various techniques, models, and evaluation methods employed in Recommender Systems. This process has deepened my understanding of state-of-the-art approaches and highlighted the significance of incorporating user preferences, sentiment analysis, and graph-based structures to build effective recommendation systems.

Moreover, this project has enhanced my ability to plan future work and identify areas for improvement. It has underscored the importance of scalability and computational efficiency when implementing recommendation systems, particularly when dealing with large datasets and sentiment analysis tasks.

Armin

In this project I had the chance to conduct much research on recommendation systems. I got to learn about the most basic reconmmendation systems and from these simple methods more elaborate methods are developed. Recommendation systems method are very diverse in how they try to solve different types of problems. I this case we got the chance to dive into the problems of course recommendation systems.

In this project I had the opportunity to implement a relative simple recommendation system prototype. I have gained experience in how to approach these systems. I learned that need to be conscious of different problems that might occur with these systems; such as memory problems due to large data sets and faulty data.

Because much time was spend on literature reviews, I have gained experience in searching and evaluating different papers and articles. I had to judge their relevance to our project and the soundness of their methods and argumentation.

Finally this project has forced me to think critically about the problems of our project. I had to think about what different problems may occur in the domain we are working in and how these problems may be solved. There was alot of reflection on why our proposed method should or shouldn't work on the given problem. In this manner we were finding the limitation of our methods, but therefore also a clear view on when to use it.

Giovanni: Working on this project, I have significantly improved my knowledge of Recommender Systems, particularly within - but not limited to - the realm of course enrolment. This happened both through the literature review - which has at times been one of my main tasks - that has given me a deeper insight into the various techniques, models, and evaluation methods employed in Recommender Systems and its state-of-the-art approaches, and through the planning and preparation of the project proposal. The preparation of the proposal has been a valuable exercise, giving a first experience at planning and organizing a project that is bigger than the ones we are used to as students, and that I guess is more in line with a Master's Thesis Project. In addition, this project has made even clearer the need to always plan accurately the future tasks, identify what could go wrong and plan for it, and which areas might need improvements.

Alex: This project improved my knowledge in recommender systems. As we were required to read various literature to propose something novel, I believe that my expertise in this field has substantially grown. In addition, I learned how to implement popular recommendation based algorithms such as user-user Collaborative Filtering. Furthermore, I got hands-on experience in working with graph data structure algorithms such as PageRank. Lastly, this project challanged me to think outside the box. While looking for a novel topic is a challanging, it also pushed me to outside of my comfort zone.