# Recommendation System for Engineering Programs Candidates

Bruno Mota da Silva
Instituto Superior Técnico
brunomotadasilva@tecnico.ulisboa.pt

Cláudia Antunes
Instituto Superior Técnico
claudia.antunes@tecnico.ulisboa.pt

## ABSTRACT

Automatic discovery of information in educational data has been broadening its horizons, opening new opportunities to its application. An open wide area to explore is the recommendation of undergraduate programs to high school students. However, traditional recommendation systems, based on collaborative filtering, require the existence of both a large number of items and users, which in this context are too small to guarantee reasonable levels of performance.

In this paper, we propose a hybrid approach, combining collaborative filtering and a content-based architecture, while exploring the hierarchical information about programs organization. This information is extracted from courses programs, through natural language processing, and since programs share some courses, we are able to present recommendations, not just based on the performance of students, but also on their interests and results in each of the courses that compose each program.

## Keywords

Recommendation systems, higher education programs, educational data mining

## 1. INTRODUCTION

Nowadays, it is common to have teenagers applying to a higher education program after finishing their high school. Every year, new programs appear and thousands of candidates must choose which one is the best for them.

This type of problem is very well-known in Educational Data Mining and in Recommendation Systems community [3, 11]. This past decade, many studies were made on creating engines that help students in choosing the courses that are suited for them, using different approaches, like content-based or collaborative filtering recommendation systems. The last type is the most used due to the large amount of data community can give.

Despite courses recommendation being a more studied problem, we want to apply these systems to programs recommendation that is not very researched yet. This brings an important challenge, since courses recommenders have already the target user inside the system rating previous courses among the others students, and in our problem candidates did not rate anything to be compared to other users in first hand.

Considering all of these aspects, our work aims for creating a recommendation system that will receive candidates personal data and high-school academic records, with the proper consent given by them considering general data protection regulations (GDPR), and will output the programs that most fit to their profile, comparing to the current student community. The system will consider the personal characteristics of the students as a matching measure and the programs' courses, objectives and description to find keywords that define the corresponding programs. These keywords will allow to compute ratings for every program considering the academic marks of the students on their own program.

This paper is divided in four more sections. Literature review covers the basic aspects of recommendation systems, with special focus on their use for educational purposes. After this, we present the architecture of our system that can be applied at a common university structure. After system architecture, current results are shown, followed by the reached conclusions at this time.

## 2. LITERATURE REVIEW

Recommendation Systems (RS) are software tools and techniques that provide suggestions for items to be of use to a user [10]. A RS can be exploited for different purposes, such as, to increase the number of items sold, to better understand what the user wants or, in another point of view, to recommend a specific item to that user.

There are two main types of recommendation engines, Content-based and Collaborative Filtering. The first one is focused on item similarities, and the second one use past behaviors of users to recommend items to the active user [1].

There is also a third type of recommendation systems, knowledge-based approaches where recommendations are given based on explicit specification of the kind of content the user wants. These systems are very similar to content-based ones, but with domain knowledge input. Finally, a hybrid recommendation system is constructed if there is a combination of two or more RS philosophies in order to improve the global performance.

Over the years, a large amount of educational data is being generated and there are being applied more collaborative filtering approaches than content-based methods in this area.

Morsomme and Alferez proposed a collaborative recommendation system that outputs courses to the target users, by exploiting courses that other similar students had taken, through k-means clustering and K-nearest neighbors techniques [2].

A recommendation system for course selection was developed in Liberal Arts bachelor of the University College Maastricht [6], using two types of data, students and courses. Student data consisted of anonymized students' course enrollments, and course data consisted of catalogues with descriptions of all courses, which allowed to find the topics of each one, using the Latent Dirichlet Allocation statistical model. Recurring to regression models of student data, the authors could predict his grade for each course. In the end, the system outputs 20 courses whose content best matches the user's academic interest in terms of Kullback-Leibler distance.

This content-based approach was applied as well in Dublin [8], where the authors used an information retrieval algorithm to compute course-course similarities, based on the text description and learning outcomes of each one.

In Faculty of Engineering of the University of Porto, it was created an engine to help students choosing an adequate higher education program to access a specific job in the future [5]. Therefore, it was implemented a recommendation system that uses the data from alumni and job offers and outputs a ranking of programs that could lead to the candidates' desired careers. The collaborative filtering approach can match the skills needed for that job and the skills given to the students of a specific degree.

Fábio Carballo made an engine that predicts students masters courses marks, using collaborative filtering methods, singular value decomposition (SVD) and as-soon-as-possible (ASAP) classifiers. With his work, he could recommend the more suitable program for students skills [4].

The topic around course and programs recommendations gained even more attention recently, with several published studies in the last years, following a variety of approaches [13, 14, 7, 9, 12].

## 3. RECOMMENDATION SYSTEM

The proposed system shall enlighten candidates about the degrees that are more compatible with their interests and that were successfully concluded by similar students, using a hybrid approach.

Our system must recommend higher education programs to a specific high-school student who wants to enroll at university. Usually, the candidate searches information about each program at universities web pages, such as courses or professional careers, or talks with students who are already enrolled at the programs he or she likes. The process of choosing a degree is very important to a high school student and it must be done analysing all the information available. Therefore, the main use case of our system focuses on candidates point of view.

As we can see on Figure 1, when the candidate uses our system, he or she must be able to give personal data that will be considered during the recommendation process. After that, the system must output a ranking of the programs that are most suitable to the candidate. Candidate's personal data can be academic interests, high school grades, personal data, such age or gender, among others. Since we are collecting data, it must be made according to the GDPR, applying anonymization techniques when necessary.

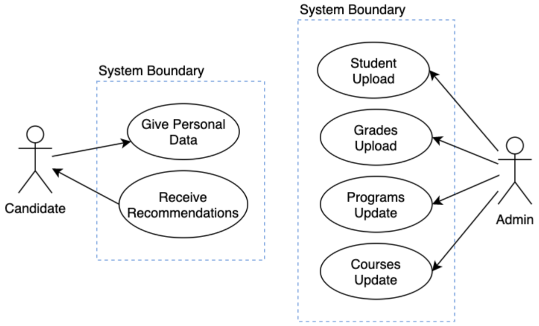

**Figure 1: Use cases of the system.**

Looking at the system from Admin point of view, there are several tasks he or she must be able to do, as we can see on Figure 1. System Administrator is the one responsible for system updates: upload new students data every year, upload students grades at the end of each semester, and update programs and courses when necessary. All the essential data to relate the candidate to current students and to make proper recommendations must be inputted before the system launching.

Finally, analysts staff can use this system when useful, to get a summary of student community and a characterization of new students.

### 3.1 Architecture

The overview of our system architecture can be seen on Figure 2, where we can distinguish two main modules: Students Profiler and Programs Recommender.

Candidates start using our system by inputting their personal data that will be used to find their profile. Current students data allow us to compute candidate profiles that will feed the second module. Programs Recommender uses the previous output to estimate a program success measure considering estimated grades, returning in the end a ranking of the most suitable programs to the candidates.

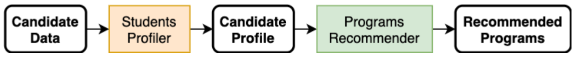

**Figure 2: Proposed architecture.**

### 3.1.1 *Students Profiler*

There is a major difference between our recommendation system and the common ones, where the target user is inside the system among the others. Here, the target user candidate is not in the system, since he or she is not enrolled at a higher education degree yet, and therefore can not rate programs or take courses. Hence, it must be developed a strategy where we can compare users.

Students Profiler computes the candidate profile as if he or she was inside the system, by comparing him with current students using shares personal variables. Therefore, the first step was to collect these data and to build a students profiling model, where we performed a feature engineering study.

A simple choice to implement Students Profiler is to apply the K Nearest Neighbors (KNN) method, after choosing the best similarity measure and number of neighbors, $K$. To compute the similarity, we used five different measures which results were studied. We also tuned the KNN process as well by trying to find the optimal value for K, that was the one having the minimum error rate.

In the end, Students Profiler returned candidate profile that will be fed to the next module.

### 3.1.2 *Programs Recommender*

Programs Recommender module is a more complex one which aims at finding the ranking of the best programs to the candidates, considering their profile **and interests**.

In order to reach its goal, this module has to create two models. The first one, called Grades Model, for estimating the candidate performance in each possible academic units, and the second one, the Ranking Model, for mapping students to programs.

As usual, the Grades Model is constructed by following a collaborative filtering approach, meaning that it uses a singular value decomposition (SVD) matrix factorization. This factorization performs a feature extraction step, reducing the number of elements to the minimum required for estimating students grades. When in the presence of the candidate profile, the Grades Model is applied to estimate the candidate grades. Using the candidate profile, instead of its original data, is the first difference in our approach, but there is more, achieved through the use of a content-based approach.

RS usually deal with a very large number of items, but the number of programs available in any university is just a few, when compared. Additionally, each student is enrolled on just one program, which means that our grades matrix would be very sparse, not contributing for a good recommendation. A third aspect is that programs share some courses (for example all engineering students study Physics and Maths, while all art students study Drawing and Geometry). But we can go a step further, and understand that courses cover some topics present in different areas. For example, several engineering courses study systems, their architecture and their dynamics.

The third proposal is the possibility of dealing with the academic units at different levels of granularity: we can aggregate everything to recommend programs, or we can simply identify a ranking of topics that are recommend for the candidate. This ability is very important to reach a new level of explainability, so needed in the field.

## 4. PRELIMINARY RESULTS

A recommendation system validation is a hard task to take. In contexts, like education, where these systems can not be made available before being proved 'correct', this task is even harder.

In our case, we made use of students data collected at the time of their enrollment in the university, to mimetize candidates surveys. Then, we used students data from 2014 to 2018 for training and data from 2019 for evaluation purposes. Moreover, every model of the system has to be validated independently, in order to better estimate each component performance, and only after tuning each of them evaluate its global quality.

We started by evaluating the Students Module, which has the use of KNN to estimate candidate profile on its basis. As data sources for this phase, we had personal data from 7918 students and grades from 7302 students, that resulted in a dataset of 7300 instances by intersecting the first ones. This dataset is composed by 101 variables, where enrolled program is the only categorical one, all of the others are numeric. Note that, we had no missing values on the dataset.

In this module, we wanted to find the K students that are most similar to the candidate. Therefore, we made a study to find the best pair (K, similarity measure) mimetizing a KNN performance study, but without focusing on the classification task. First, we needed to define which condition must students achieve to have success on their program, based on their Grade Point Average (GPA), from a 0-20 scale. Hence, a histogram was made and it is shown at Figure 3.

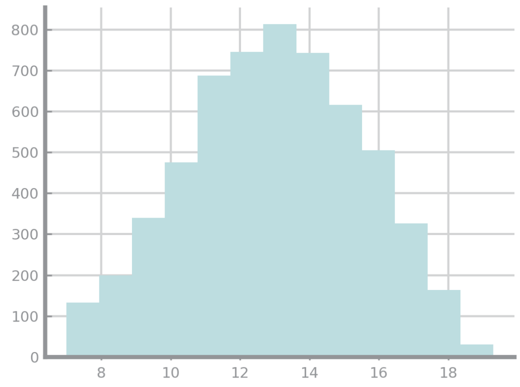

**Figure 3: Number of students by each GPA class.**

Since the average of students GPA is 12.99, we labeled as having success students which GPA was equal to 13 or more, and not having success otherwise. This way we guaranteed a balanced dataset. After the labelling, we computed ten trials of data train-test split for five similarity measures (chebyshev, correlation, cosine, euclidean, and manhattan) and for K between 5 and 155 in multiples of 5. For each pair (K, similarity measure), we computed the average of KNN model accuracies, since 70% train and 30% test datasets are random in each trial. The results are shown in Figure 4, and zoomed in Figure 5.

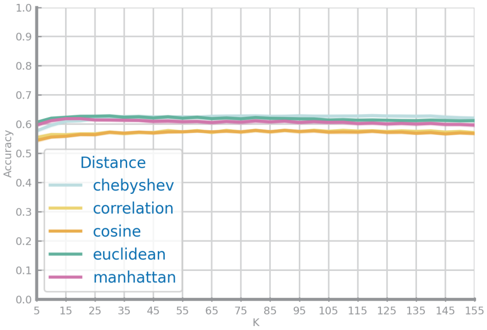

**Figure 4: K and similarity measure study.**

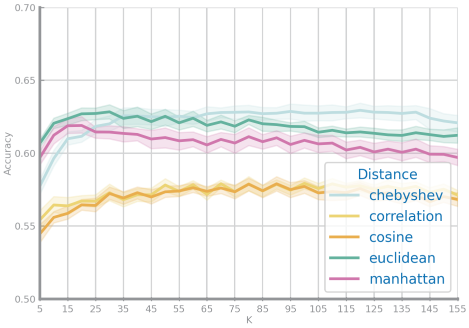

**Figure 5: Zoom of K and similarity measure study.**

Students Profiler module has five conditions that will be tested in the global system: (120, chebyshev); (100, correlation); (90, cosine); (30, euclidean) and (20, manhattan).

We implemented as well a simple recommendation system where we used the candidate profile, composed by the average grades of all neighbors for all courses taken by them, to predict the candidate grades for all available courses. In this component, four conditions were used for testing the system behaviour for all similarity measures: A) using SVD as matrix factorization technique with the K values mentioned above and considering all the variables from students data; B) same as A), but using K equals to 5; C) using SVD with the best K values predicted using a reduced students dataset with only academic records; and D) same as C) but using the Slope One prediction method.

After that, we used 1509 candidates to test the system, where we computed the GPA that each of them would have in each one of the available programs using their predicted course grades and ranked them by GPAs. Then, we computed the mean absolute error for those which first recommended program coincides with their current program in terms of GPA, and results are showed in Table 1.

**Table 1: Mean Absolute Errors for each prediction method and for each similarity measure**

| Similarity Measure | A | B | C | D |
|---|---|---|---|---|
| chebyshev | 2.065 | 2.378 | 2.139 | 2.289 |
| correlation | 2.149 | 2.580 | 2.359 | 2.325 |
| cosine | 2.153 | 2.583 | 2.430 | 2.451 |
| euclidean | 2.538 | 2.497 | 2.153 | 2.289 |
| manhattan | 2.313 | 2.488 | 2.376 | 2.451 |

The next steps will consist of improving the way we recommend the programs and its ranking model, considering different ensembles, namely random forests and gradient boosting. At this time, we are predicting GPA with almost 90% accuracy.

# 5. CONCLUSIONS

The current educational context, even more after the beginning of the pandemic situation, demands new educational systems. Systems able to address the difficulties inherent to distance learning contexts, where students are far from educators, and plenty of times try to follow their path without any guidance. Most of the times, online education tools deal with students in a 'one-fit-all' approach, that ignore each students preferences.

In this paper, we propose a new architecture for a recommendation system, designed for suggesting programs to university candidates. Our system benefits from an hybrid architecture, that combines collaborative filtering with a content-based philosophy, exploring the full documentation of programs and courses available. Additionally, we explored the notion of feature stores to easily update the data repositories to support our system.

The proposed architecture is adaptable to smaller contexts, for example for suggesting learning resources at any abstraction levels, such as exercises.

# 6. ACKNOWLEDGMENTS

This work was supported by national funds by Fundação para a Ciência e Tecnologia (FCT) through project GameCourse (PTDC/CCI-CIF/30754/2017).

# 7. REFERENCES

[1] C. C. Aggarwal. *Recommender Systems: The Textbook*. Springer Publishing Company, Incorporated, 1st edition, 2016.

[2] A. Al-Badarneh and J. Alsakran. An automated recommender system for course selection. *International Journal of Advanced Computer Science and Applications*, 7, 03 2016.

[3] R. S. Baker and K. Yacef. The state of educational data mining in 2009: A review and future visions. *Journal of Educational Data Mining*, 1(1):3–17, 2009.

[4] F. O. G. Carballo. Masters' courses recommendation: Exploring collaborative filtering and singular value

decomposition with student profiling. 2014.

[5] A. I. N. A. de Sousa. Market-based higher education course recommendation. 2016.

[6] R. Morsomme and S. V. Alferez. Content-based course recommender system for liberal arts education. 2019.

[7] B. MS, Y. Taniguchi, and S. Konomi. Course recommendation for university environments. 07 2020.

[8] M. P. O'Mahony and B. Smyth. A recommender system for on-line course enrolment: An initial study. page 133–136, 2007.

[9] A. Polyzou, A. N. Nikolakopoulos, and G. Karypis. Scholars walk: A markov chain framework for course recommendation. 05 2019.

[10] F. Ricci, L. Rokach, and B. Shapira. Recommender systems handbook. *Recommender Systems Handbook*, 1-35:1–35, 10 2010.

[11] A. Rivera, M. Tapia-Leon, and S. Luján-Mora. Recommendation systems in education: A systematic mapping study. pages 937–947, 01 2018.

[12] F. Scherzinger, A. Singla, V. Wolf, and M. Backenköhler. Data-driven approach towards a personalized curriculum. 07 2018.

[13] R. Yu, Q. Li, C. Fischer, S. Doroudi, and D. Xu. Towards accurate and fair prediction of college success: Evaluating different sources of student data. 07 2020.

[14] Y. Zhao, Q. Xu, M. Chen, and G. M. Weiss. Predicting student performance in a master of data science program using admissions data. 07 2020.