An Online Course Recommendation System for Career Paths

Maria Panagou

December 2020




ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Athens University of Economics and Business

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

School of Information Sciences and Technology
Department of Informatics
Athens, Greece

Master Thesis
in
Computer Science

An Online Course Recommendation System for Career Paths

Maria Panagou

Committee: Professor Vasiliki Kalogeraki (Supervisor)

Associate Professor Iordanis Koutsopoulos

Assistant Professor Spyridon Voulgaris

November 2020



Maria Panagou

An Online Course Recommendation System for Career Paths

November 2020

Supervisor: Prof. Vasiliki Kalogeraki

Athens University of Economics and Business

School of Information Sciences and Technology

Department of Informatics

Mobile Multimedia Laboratory

Athens, Greece



Abstract

Online courses have become extremely popular these days, as more and more universities, institutes and companies are offering online courses available for anyone. These courses provide an affordable and flexible way to learn new skills, advance your career and deliver quality educational experiences at scale. Millions of people around the world enroll to online courses to learn for a variety of reasons including: career development, changing career, college preparation, supplemental learning, lifelong learning, corporate eLearning training, and more. However, despite the popularity of online education, vast groups of people consciously stay away from such methods, mostly due to the wide variety of courses and the growing number of available online courses.

In this context, recommendation systems can play a decisive role in making better and wiser decisions that can affect someone's future. In this thesis, a proof-of-concept of an online course recommendation system is proposed that aims to improve users' career readiness by suggesting relevant skills and courses based on their unique career interests.


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Περίληψη

Τα διαδικτυακά μαθήματα έχουν γίνει εξαιρετικά δημοφιλή αυτές τις μέρες, καθώς όλο και περισσότερα Πανεπιστήμια και εταιρείες προσφέρουν online μαθήματα. Αυτά τα μαθήματα παρέχουν έναν προσιτό και ευέλικτο τρόπο για τον οποιονδήποτε να μάθει νέες δεξιότητες και να εξελίξει την καριέρα του. Εκατομμύρια άνθρωποι σε όλο τον κόσμο εγγράφονται σε διαδικτυακά μαθήματα για διάφορους λόγους, όπως για: εξέλιξη ή αλλαγή σταδιοδρομίας, προετοιμασία για το Πανεπιστήμιο, δια βίου μάθηση και πολλά άλλα. Ωστόσο, παρά τη δημοτικότητα της διαδικτυακής εκπαίδευσης, πολλοί είναι αυτοί που απέχουν συνειδητά, κυρίως λόγω της μεγάλης ποικιλίας των μαθημάτων και του όλο και αυξανόμενου αριθμού διαθέσιμων διαδικτυακών μαθημάτων.

Προς αυτήν την κατεύθυνση, τα recommendation systems μπορούν να παίξουν καθοριστικό ρόλο καθώς μπορούν να βοηθήσουν στη λήψη καλύτερων αποφάσεων που μπορεί να επηρεάσουν το μέλλον κάποιου. Σε αυτή τη διπλωματική εργασία, προτείνεται ένα recommendation system για διαδικτυακά μαθήματα που στοχεύει στη βελτίωση των προσόντων των χρηστών προτείνοντας σχετικές δεξιότητες και μαθήματα με βάση τα επαγγελματικά τους ενδιαφέροντα. Η κύρια ιδέα είναι ότι ο χρήστης μπορεί να εισάγει την εργασία που θέλει να ακολουθήσει και στη συνέχεια του προτείνονται μαθήματα με βάση τις δεξιότητες που απαιτούνται για την εργασία που έχει επιλέξει. Αν ο χρήστης διαθέτει κάποιες από αυτές τις δεξιότητες, τότε μπορεί απλά να τις αφαιρέσει και τα προτεινόμενα μαθήματα προσαρμόζονται ανάλογα με τη καινούρια του αναζήτηση.

Αρχικά, φτιάξαμε ένα web scaper για να συλλέξουμε δεδομένα από το Coursera, μία πλατφόρμα με διαθέσιμα διαδικτυακά μαθήματα, και συγκεκριμένα πήραμε τους τίτλους των μαθημάτων, το Πανεπιστήμιο που τα παρέχει, τη βαθμολογία τους, τον αριθμό των χρηστών που έκαναν την αξιολόγηση, τον αριθμό των μαθητών που έχουν παρακολουθήσει το κάθε μάθημα και τις δεξιότητες που αποκτάς. Στη συνέχεια, φτιάξαμε ακόμα ένα web scraper για να συλλέξουμε ένα δεύτερο σύνολο δεδομένων από το Zippia, ένα site για εύρεση εργασίας, με τα προσόντα που απαιτούνται για την κάθε εργασία. Στην τελική φάση, για να ολοκληρώσουμε το recommendation system, χρησιμοποιήσαμε *cosine similarity* και *Jaccard similarity* για να προτείνουμε μαθήματα και συγκρίναμε τα αποτελέσματα.


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Acknowledgement

I would like to thank two important groups of people without whom this thesis would not have been possible: my committee and my family.

Firstly, the members of my thesis committee for their intellectual contributions to my development as a computer scientist. In particular, to my thesis advisor, Dr. Vasiliki Kalogeraki, for her guidance throughout the research process and for being there whenever I ran into a problem in my research to help me overcome any issues. And, to my committee members, Dr. Iordanis Koutsopoulos and Dr. Spyridon Voulgaris, for the appreciation and excitement inspired by their lectures.

Finally, I wish to express my most sincere gratitude and appreciation to my husband, for the support he provided me through these years. Without his love, encouragement, and his editing assistance, I would not have finished this thesis. And, also, to thank my parents for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Contents

Abstract vii

Acknowledgements ix

1 Introduction 1

1.1 Thesis Structure . . . 2

2 Background and Related Work 5

2.1 Background on Recommender Systems . . . 5

2.1.1 Content-Based Filtering . . . 5

2.1.2 Collaborative Filtering . . . 8

2.1.3 Collaborative filtering: Memory-Based Approach . . . 9

2.1.4 Collaborative filtering: Model-Based Approach . . . 11

2.2 Related Work . . . 11

2.2.1 Online Courses Recommendation based on LDA . . . 11

2.2.2 MoocRec.com : Massive Open Online Courses Recommender System 12

2.2.3 A Recommendation System for Online Courses . . . 12

2.2.4 Online courses recommendation system based on industry occupation skills requirements . . . 12

2.2.5 CaPaR: A Career Path Recommendation Framework . . . 13

2.2.6 Towards an Intelligent Hybrid Recommendation System for E-Learning Platforms Using Data Mining . . . 13

2.2.7 A Collaborative Recommendation System for Online Courses Recommendations . . . 13

2.2.8 Recommendation System for Student E-Learning Courses . . . 14

3 System Design and Implementation 15

3.1 System Architecture . . . 15

3.2 Implementation . . . 15

3.2.1 Web Scraping for Data Selection . . . 17

3.2.2 Data Pre-processing . . . 18

3.2.3 Recommendation Engine . . . 19

3.3 Design . . . 20

3.3.1 Technical Stack . . . 21

3.3.2 Algorithm . . . 21

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

4 Evaluation **23**
4.1 Data Selection and Visualization 23
4.2 Evaluation Metrics 24
4.3 Experiment 26
4.3.1 Experimental Setup 26
4.4 Evaluation approach 27
4.5 Results 28
4.5.1 Experiment Case Studies 30

5 Conclusions **39**

Bibliography **41**

List of Acronyms **43**

List of Figures **45**

List of Tables **47**

List of Algorithms **49**

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

1 Introduction

Using recommender systems to filter vast amounts of information has been done successfully in a variety of areas and are most commonly recognized as playlist generators for video and music services like Netflix [1], YouTube [2] and Spotify [3], product recommenders for services such as Amazon [4], or content recommenders for social media platforms such as Facebook [5] and Twitter [6]. These systems can operate using a single input, like music, or multiple inputs within and across platforms like news, books, and search queries. There are also popular recommender systems for specific topics like restaurants and online dating. Recommender systems have also been developed to explore research articles and experts, collaborators, and financial services [Wik20d]. We wanted to deploy those same techniques used to recommend movies to make a personalized recommender system for online courses.

An issue that we face nowadays, is that many people don't have the opportunity to study due to economical or other factors. In addition, students who graduated from universities do not always have the necessary skills that the company employer requires. This issue is also known as the gap between industry and education. In this context, educational needs vary from people to people based on their career goals and skills-gap. In this direction, online courses play a tremendous role. But the process of finding appropriate courses and deciding for the right one, can be challenging and time-consuming. There are many online platforms available with an enormous range of courses, and many factors to consider, such as the skills gained of the course, the cost, the level, the length and the prerequisites. Trying to skip this difficulties by using other people's experience and suggestions to choose a course may be misleading, because each person has their own background, education, and desired career goals. So, a course that was helpful for one person might be somehow unrelated to another's career interests.

Almost none of the currently used course recommendation systems or online platforms take into consideration the user's future career goal combined with the qualifications that already has. Instead, they suggest courses based either on what skills the user wants to

[1] https://www.netflix.com/
[2] https://www.youtube.com/
[3] https://www.spotify.com/gr/
[4] https://www.amazon.com/
[5] https://www.facebook.com/
[6] https://twitter.com/

gain or the career path she wants to follow. For instance, Linkedin [8] proposes learning paths based on users career interests. In this thesis, we are proposing a personalized online course recommendation system based on career paths and also, on the skills that the user already posses, aiming to improve the recommendation results. Our approach offers several advantages over previous recommender systems:

1. **The recommendations are relevant:** Our recommendations rely on features of objects themselves rather than other users with similar interests. So, they are likely to be highly relevant to the user's unique job interests.

2. **New items can be recommended immediately:** Our algorithm does not require a user to interact with an item before it can be recommended, so new items can be recommended immediately.

3. **Avoids cold-start problem:** This approach helps users to get started more quickly, because a user must enter search terms in order to get recommendations.

4. **Multiple benefits for people involved:** The stakeholders that have multiple benefits are:

    - **Users:** Users can discover both the skills and the courses needed to obtain the job of their interest.
    - **Employers:** Learners enter the work industrial with the skills needed to make an immediate impact.

1.1 Thesis Structure

This thesis is organized as follows:

In **Chapter 2:**

We provide the background on the topic of recommender systems, we analyze their techniques and we quote the advantages and disadvantages of them. Furthermore, we discuss related works using recommendation systems, especially in the field of course recommendation systems.

In **Chapter 3:**

We introduce the architecture of our system. We analyze the design and the different parts that contains, the technical implementation, such as the different libraries and frameworks that we used, and we quote our algorithm.

[8] https://www.linkedin.com/

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

In **Chapter 4:**

We describe the results of our experiment. In addition, we analyze further some experiments with certain criteria and we evaluate our recommender system according to the evaluation metrics that we chose. These metrics are presicion, recall and F-Score.

Finally, in **Chapter 5:**

We discuss the conclusions and present possible future work. More specifically, we present briefly our system, its advantages and disadvantages and suggest things to do that could improve our system in the future.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Background and Related Work

2

2.1 Background on Recommender Systems

Recommender systems can be classified into two basic architectures: *content-based filtering* and *collaborative filtering*.

- **Content-Based Filtering:** Content-based systems focus on properties of items. Similarity of items is determined by measuring the similarity in their properties.
- **Collaborative Filtering:** Collaborative-filtering systems focus on the relationship between users and items. Similarity of items is determined by the similarity of the ratings of those items by the users who have rated both items.

Hybrid recommender systems combine the two basic approaches.

2.1.1 Content-Based Filtering

Content-based filtering is based on a description of the item and a profile of the user's preferences. These methods are best suited to situations where there is known data on an item (name, location, description, etc.), but not on the user. Content-based recommenders treat recommendation as a user-specific classification problem and learn a classifier for the user's likes and dislikes based on an item's features [Das15]. Most text mining techniques cannot process text directly to extract item's features. Instead, the text needs to be transformed to values representing characteristics of the text known as a feature vector. This process is known a vectorization (Word Embeddings).

Vector space model is an algebraic model, involving two steps. In first step the text document is been represented into vector of words and in second step is been transformed to numerical format so that text mining techniques such as information retrieval, information extraction,information filtering etc. can be applied. This model, also, measures the degree of similarity between the vectors. As each item is stored as a vector of its attributes, the angles between the vectors are calculated to determine the similarity between the

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

vectors. The smaller the angle, the more similar the vectors. This is illustrated in Figure 2.1 [Wik20e].

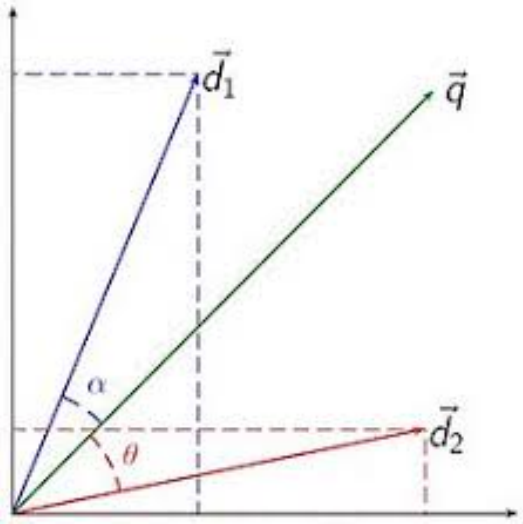


Fig. 2.1: Vector Space Model

A widely used technique for extracting features from items is **Term Frequency - Inverse Document Frequency (TF-IDF)**

TF: Term frequency. This is simply the frequency of a word in a document.

IDF: Inverse Document Frequency . This is the universe of document frequency among the whole corpus of documents.

Each feature in the item and user profile vectors is represented by its TF-IDF weight. The advantage of TF-IDF weighting is that negates the effect of high frequency words in determining the importance of an item (document).

Count Vectorization is another technique for word embeddings. The idea is to collect set of words, sentences or documents and count occurrence of each word in document. It is also called as one-hot encoding.

Once the features of the items are been extracted, the next step is to measure the similarity between them. There are many techniques to do that, such as *Euclidean Distance, Pearson's Correlation Coefficient, Cosine Similarity, Jaccard Similarity Coefficient* and *Tanimoto Coefficient.* In this thesis, we used two of the most popular: **Cosine Similarity** and **Jaccard Coefficient**.

Cosine similarity is a metric for computing similarities between two vectors in the vector space model, by measuring the cosine angle between vectors. The items with the highest

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

cosine similarity values are returned as recommendations. For each item, it takes the dot product of the item and user profile vectors using the formula:

$$Cosine(A, B) = \frac{A \cdot B}{||A|| \cdot ||B||} \tag{2.1}$$

Another way to measure similarity between documents is **Jaccard Similarity Coefficient** [Wik20b] which is also known as Intersection over Union. The Jaccard coefficient is a statistic used for gauging the similarity and diversity of sample sets. In practice, measures similarity between finite sample sets, and is defined as the size of the intersection divided by the size of the union of the sample sets, as shown in Figure 2.2:

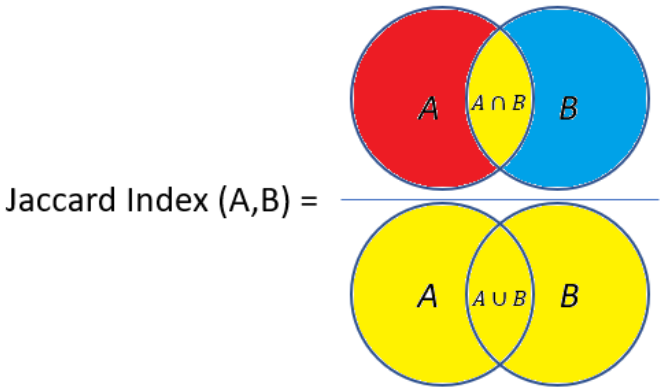


Fig. 2.2: Jaccard similarity

Jaccard's similarity formula is shown in Equation 2.2:

$$Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \tag{2.2}$$

Major difference between Jaccard and cosine similarity is that Jaccard similarity takes set of unique length of words instead cosine similarity takes whole sentence vector. So, if data duplication is not matter then its better to use Jaccard similarity, else cosine similarity is good for measuring the similarity between two vectors even if the data duplication is there.

Advantages and Disadvantages of Content-Based Filtering

A major advantage of the content-based filtering approach is that recommendations are specific to users' unique interests and does not need any information about other users. This allows content-based recommender systems to avoid the cold-start problem for new

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

and unpopular items. Other advantages are transparency and that make users tend to like and feel more confident about recommendations.

There can be, also, disadvantages in using content-based filtering. In particular, the method provides over-specialization and a limit degree of novelty, since it has to match up the features of profile and items. A totally perfect content-based filtering may suggest nothing "new" and cannot recommend items outside of a user's preferences. Furthermore, it is susceptible to the cold-start problem for a new user, when there's not enough information to build a solid profile. Lastly, content-based filtering cannot take advantage of quality assessments made by other users.

2.1.2 Collaborative Filtering

The motivation for collaborative filtering comes from the idea that people often get the best recommendations from someone with tastes similar to themselves. Collaborative filtering encompasses techniques for matching people with similar interests and making recommendations on this basis. This paradigm is based on the assumption that users with similar interests in the past will have similar interests in the future, like in Figure 2.3.

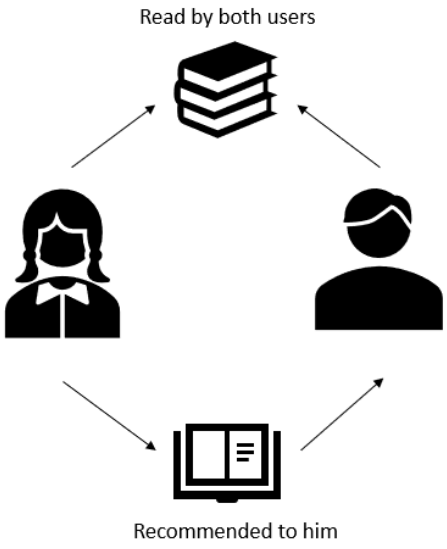


Fig. 2.3: Collaborative Filtering

The algorithm follows the following steps:

- A user expresses her preferences by rating items. These ratings can be viewed as an approximate representation of the user's interest in a particular domain.
- The system matches user's ratings against other users and finds the people with most "similar" tastes.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

- With similar users, the system recommends items that the similar users have rated highly but not yet being rated by this user.

Two major approaches of collaborative filtering algorithms exist today: **memory-based** and **model-based**.

2.1.3 Collaborative filtering: Memory-Based Approach

Memory-based collaborative filtering algorithms use the entire database to generate a recommendation by using user rating data to compute the similarity between users or items. These algorithms include two steps: first, they find a set of users or items that are similar to the active user and have rated an item. Second step is to calculate the rating of the active user based on the ratings of the similar users.

Memory-Based Collaborative Filtering approaches can be further divided into two main sections: **user-item filtering** and **item-item filtering**, as shown in Figure 2.4.

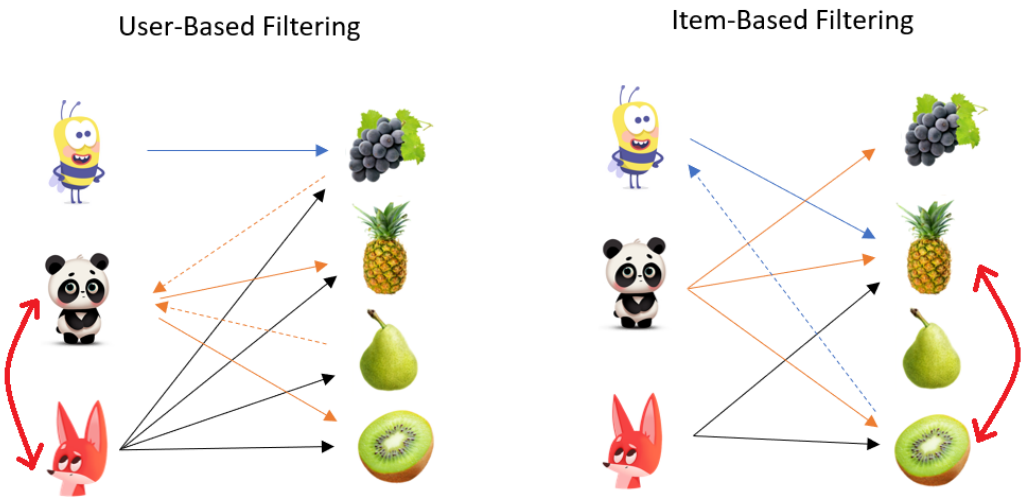


Fig. 2.4: Memory-based Techniques

User-Item filtering: This approach will take a particular user, finds users that are similar to that user based on similarity of ratings, and recommends items that those similar users liked, i.e.: *"Users who are similar to you also liked ..."*. A specific application of this is the user-based Nearest Neighbor algorithm. This algorithm needs two tasks:

1. Find the K-nearest neighbors (k-NN) to the user a, using a similarity function w to measure the distance between each pair of users (Equation 2.3).

$$Similarity(a, i) = w(a, i), i \in K \tag{2.3}$$

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

2. Predict the rating that user a will give to all items the k neighbors have consumed but a has not.

The four commonly used similarity measures to quantify distance between two users are **Euclidean distance, Pearson's correlation coefficient, Jaccard coefficient** and **cosine similarity** [Sha19]. Once the top k neighbors have been identified, a user-item matrix, such as the one shown in Figure 2.5, is used to recommend items that the active user had never seen, based on the ratings that are provided by the k neighbors.

	Item A	Item B	Item C
User A	4	2	?
User B	5	?	5
User C	1	?	3

Fig. 2.5: User-Item Matrix based on k Neighbors Ratings

Item-Item filtering: In contrast, this approach will take an item, find users who liked that item, and find other items that those users or similar users also liked. It takes items and outputs other items as recommendations, i.e.: *"Users who liked this item also liked …"* [Spa20].

Like user-based collaborative filtering, similarity between two items can be calculated using any similarity measure. The difference between these two approaches of memory-based filtering is that, in this case, the algorithm directly pre-calculate the similarity between the co-rated items, skipping K-neighborhood search.

Advantages and Disadvantages of Memory-Based Approach

The advantages of this approach is that it is easy to implement and it is more accurate than other techniques.

On the other hand, it's drawbacks [Pin17] are:

- **Sparsity:**The percentage of people who rate items is really low.
- **Scalability:** The more k neighbors we consider (under a certain threshold), the better the classification should be. Nevertheless, the more users there are in the system, the greater the cost of finding the nearest K neighbors will be.
- **Cold-start:** New users will have little information about them to be compared with other users.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

- **New item:** New items will lack of ratings to create a solid ranking.

2.1.4 Collaborative filtering: Model-Based Approach

Model Based algorithms received a significant push in the research community after the million dollar prize competition opened by Netflix.com in 2006. In this approach, models are developed using different data mining, machine learning algorithms to predict users' rating of unrated items. Model-based recommendation systems use only a subset of information about users and items to use as a representative "model" for making recommendations. The reduced dimensionality offers benefits of both speed and scalability over the memory-based approach which uses the whole dataset. There are many model-based CF algorithms to reduce the dimensionality of a rating matrix, such as *Bayesian Networks, Clustering Models, Latent Semantic Models such as Singular Value Decomposition, Probabilistic Latent Semantic Analysis, Multiple Multiplicative Factor, Latent Dirichlet Allocation, Markov Decision Process Based Models and Probabilistic Matrix Factorization* [Wik20a].

One drawback of the model-based approach is that due to generalization, recommendations often result in lower levels of accuracy than those in memory-based algorithms.

2.2 Related Work

The need of lifelong learning has led to the rise of online courses. But enrolling to the right course sometimes become difficult due to the variety of both courses and online courses' platforms. The appropriate course requires careful consideration of a number of factors ranging from users' career interests to time commitment, level of difficulty and existing users' skills. Over the last decade, several researchers have proposed recommendation systems that incorporate collaborative filtering methods, content-based methods, or the combination of them as hybrid systems. But in the field of online courses recommendation systems the research was rather limited. In this section, some of the most relevant systems to this research are reviewed.

2.2.1 Online Courses Recommendation based on LDA

In 2014, *Rel Guzman Apaza et al.* from National University of St. Agustin of Arequipa in Peru, proposed a course recommendation system based on historical grades of students in college. Their model is able to recommend available courses in sites such as: Coursera, Udacity, Edx, etc. [Apa+14]. To do so, probabilistic topic models are used as follows. On one hand, Latent Dirichlet Allocation (LDA) topic model infers topics from content given in a college course syllabus. On the other hand, topics are also extracted from a massive

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

online open course (MOOC) syllabus. These two sets of topics and grading information are matched using a content based recommendation system so as to recommend relevant online courses to students. Preliminary results show suitability of our approach.

2.2.2 MoocRec.com : Massive Open Online Courses Recommender System

In 2016, *P. Symeonidis* and *D. Malakoudis* from Department of Informatics, of Aristotle University in Thessaloniki, proposed a web site that recommends courses to users so that, they can acquire those skills that are expected from their ideal job posting [SM16]. MoocRec's recommendation engine is based on Matrix Factorization (MF) model combined with Collaborative Filtering (CF) algorithm, which exploits information from external resources (i.e., users' skills, courses' characteristics, etc.) to predict course trends and to perform rating predictions according to them.

2.2.3 A Recommendation System for Online Courses

In 2017, *David Estrela, Sérgio Batista, Diogo Martinho* and *Goretia Marreiros* developed a recommendation engine that uses three techniques to extract the information and suggest online courses: Content-Based, Collaborative filtering and Hybrid [Est+17]. By combining these three techniques the system can offer more accurate recommendations and only considers the interests of each user for online courses. Although the particular developed system uses quite simple recommendation algorithms, one drawback of it is related to the difficulty to provide recommendations right after the user registers since there is still no information about the user, which is known as cold-start problem. So the user must provide some information to skip this problem. Another drawback is related to the size of the dataset, since it was quite small, the recommendation precision was not as accurate as expected.

2.2.4 Online courses recommendation system based on industry occupation skills requirements

In 2017, *Hai-Hui Wang et al.* from Department of Computer Science and Information Engineering National Central University Taoyuan in Taiwan, proposed a clustering method that shows the results of groups of skills that are commonly needed for a particular type of job [Wan+17] .

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

2.2.5 CaPaR: A Career Path Recommendation Framework

In 2017, *Bharat Patel, Varun Kakuste* and *Magdalini Eirinaki* from Computer Engineering Department San Jose State University, proposed a career path recommendation framework, that takes into consideration the user's profile and skills, to generate more relevant career recommendations for users [PKE17]. Using text mining and collaborative filtering techniques the system first scans the user's profile and resume, identifies the key skills of the candidate and generates personalized job recommendations. Moreover, the system recommends additional skills to students required for related job openings, as well as learning resources for each skill. In this way, the system not only allows its users to explore large amounts of information, but also expand their portfolio and resume to be able to advance their careers further.

2.2.6 Towards an Intelligent Hybrid Recommendation System for E-Learning Platforms Using Data Mining

In 2017 *Marouane El Mabrouk et al.* presented an intelligent hybrid recommendation system based on data mining [MGR17]. This system has four parts, the first for data collection and for center of interest construction by two modes: explicit data collection, which based on users and what they filled in their profiles, and implicit and automatic data collection by proposing a survey to users in order to gather information about their interests. A second part for processing information already collected in the previous part and for creating the learning model, classifying users who posted the content and classifying content also in order to send the results to the recommendation module. The third part is for making the similarity between learners and content and doing the recommendation for learners and the final part is for creating a log file of recommendation by learner, which will be used in the upcoming recommendation.

2.2.7 A Collaborative Recommendation System for Online Courses Recommendations

In 2019, *Raghad Obeidat, Rehab Duwairi* and *Ahmad Al-Aiad* from Jordan University of Science and Technology in Irbid, proposed a collaborative recommender system that recommends online courses for students based on similarities of students' course history [ODA19]. First, clustering is used to classify or group students with the same interest. When groups have been established, data mining techniques will be used to elicit the rules of the best learning path. Finally, data mining techniques have been, also,used by applying association rules algorithms to generate courses rules.



2.2.8 Recommendation System for Student E-Learning Courses

In 2019, *K.Nasaramma et al.* from Vignan's Institute Of Information Technology of Visakhapatnam in India, developed an e-learning system which recommends e-learning courses to students based on what other similar students have taken and specifies expected grades for these courses [KRK20]. This system automatically finds similar students and then they apply association rule mining algorithm (Apriori algorithm) on their courses to create courses association rules.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

3 System Design and Implementation

The architecture of our online course recommender system can be summarized into three parts. In this chapter we present the architecture and the implementation of our system, and we describe its parts.

3.1 System Architecture

The system consists of three parts:

1. **User's Input:** Accepts user query for career path and difficulty level, e.g., "Beginner", "Mixed", "Intermediate".
2. **Data Pre-Processing:** After web scraping from *Coursera* [13] the online courses titles with their features, e.g. skills gained and from *Zippia* [14] the skills required for each job specialization, normalizes both datasets.
3. **Recommendation Engine:** Extracts the skills from user's query and converts them to vectors using two methods, TF-IDF Vectorizer and CountVectorizer, under the Vector Space Model.Computes the similarity score between the required skills for user's desired career path and the skills gained from each online course using both cosine and Jaccard similarity. Recommends top N courses with highest cosine similarity score, and, also, the top N courses with the highest Jaccard similarity score.

The overall system architecture for our system is shown in Figure 3.1.

3.2 Implementation

Our recommendation system consists of two datasets, the **Jobs Database** and the **Online Courses Database**. Both datasets include a column that refers to skills, either gained

[13] https://www.coursera.org/
[14] https://www.zippia.com/

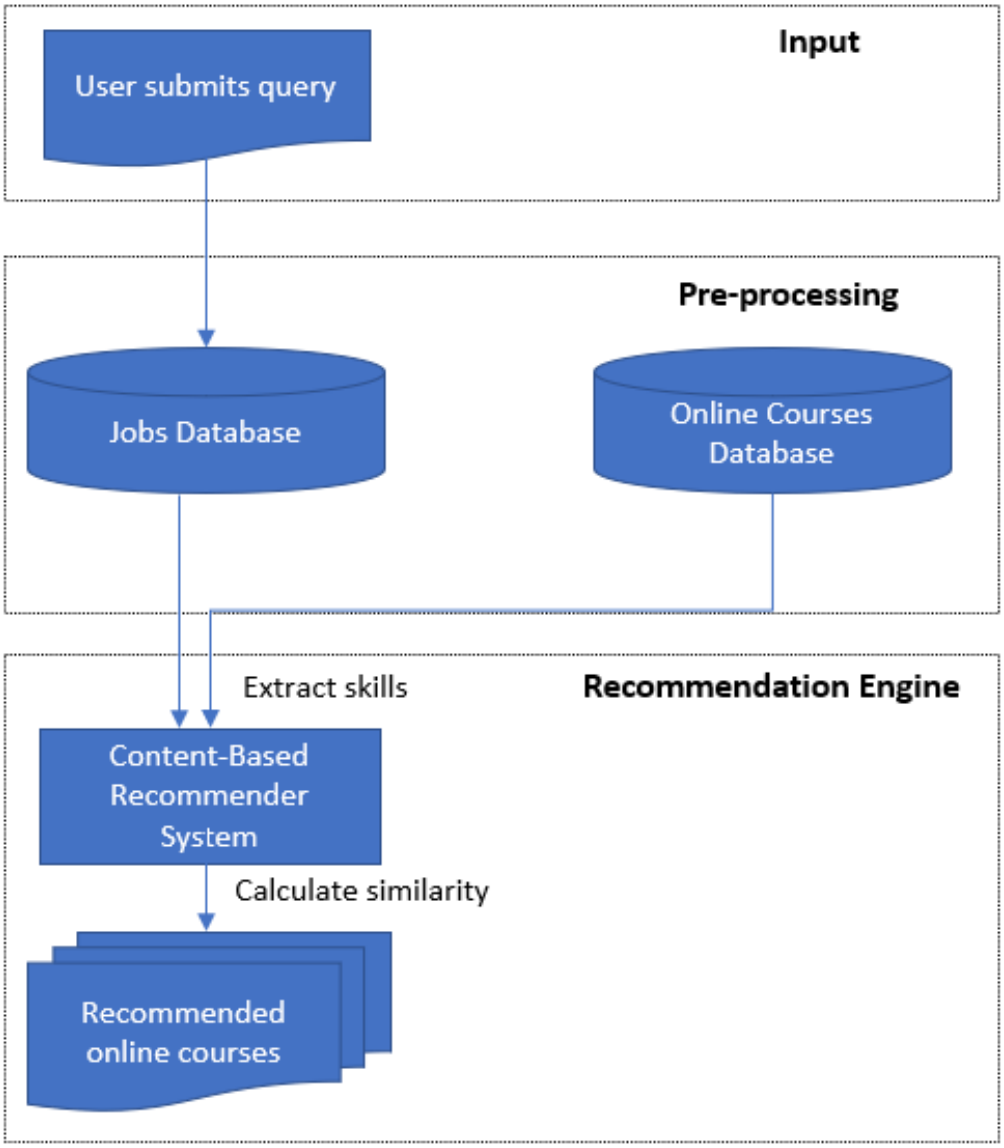


Fig. 3.1: Architecture Diagram of the Recommendation System

skills from online courses, or required skills for job specializations. The system match these skill sets in order to recommend online courses.

First, our system accepts a user's query about a job specialization of her interest and a desired difficulty level, and it uses the *Jobs Database* to generate a list of required skills based on the user's query. If the user posses any of the required skills, she can subtract these skills from the required skill-set. Once the system extracted the required skills, it is essential to apply normalization techniques to both extracted skills and to *Online Courses Database* for matching. Next step, includes the *Vector Space Model* to convert both the extracted skills-features, and the gained skills-features from the *Online Courses Database* into vectors. Finally, our system uses *Cosine and Jaccard Similarity* to compute

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

the similarity between these two features and recommends the respective top-N courses with the highest similarity score for each similarity measure.

3.2.1 Web Scraping for Data Selection

There are two sets of data that we collected with web scraping:

1. the *online course* database which contains information related to the courses such as title, rating, reviews, URL, skills and other metadata,
2. and *job specialization* database that contains jobs titles especially pertaining to roles in the field of Computer Science with their required skills.

Online Course Data: The online course database was collected from *Coursera.org* with web scraping. Each course in the course database contains:

- **Title**, the complete title of the learning product.
- **Organization**, the university or institution that provides the learning product.
- **Certificate Type**, the learning product, i.e. course, professional certificate or specialization.
- **Rating**, the feedback for each learning product.
- **Reviews**, the evaluation for each learning product.
- **Level**, the student's level, i.e. beginner, intermediate, mixed or advanced.
- **URL**, the learning product's link.
- **Enrolled Students**, the number of students that have already enrolled and
- **Skills**, what you gain after the completion of the learning product.

In total, 891 courses were collected. We removed manually 235 courses that did not contain skills and 27 courses that were offered in languages other than English. We estimated that the rating of the courses is ranging between 3.9 to 5.0 (Figure 3.2 shows the course ratings' distribution from users that have completed the courses). Due to the fact that the majority of the courses have very good rating, we removed the courses with rating between 3.9-4.2 and, finally, we were left with 620 unique courses, professional certificates, specializations and their features.

Job Specialization Data: In the absence of a public dataset of required job skills for roles in the field of Computer Science and other fields, we built a web scraper to collect job titles and their skills from *zippia.com*. We selected 27 job titles in the field of Data Science and 10 job titles of other specializations. In total, approximately we collected 14 high demand

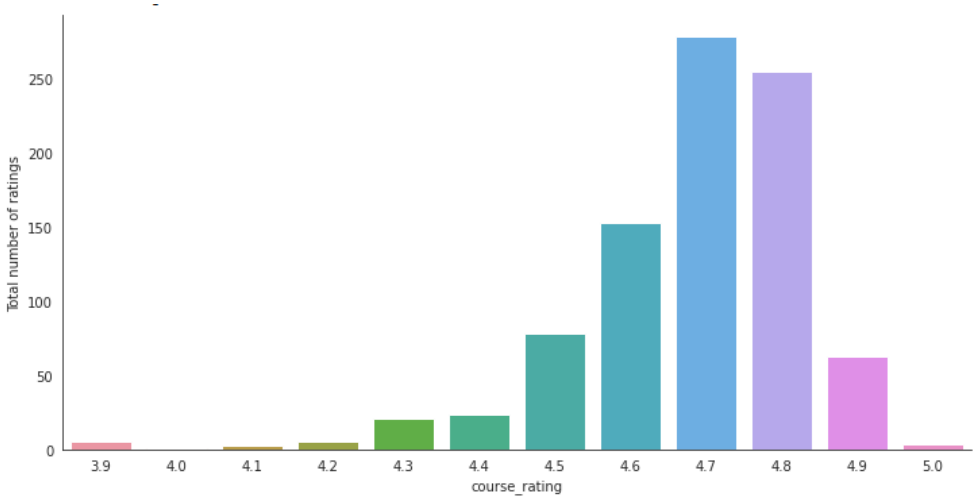


Fig. 3.2: Course Ratings' Distribution

skills for each job specialization. After removing similar skills, we were left with a total of 10 unique technical skills per specialization. For the purposes of conducting a controlled experiment, we limited our dataset to job specializations mainly in the field of Computer Science. In the future, the model can be extended to support additional fields.

3.2.2 Data Pre-processing

For parity, after extensive data exploration, the following dimensionality reduction and normalization techniques were applied to both courses and job specializations databases:

- **Removed**:
  - **Punctuation**. In particular we removed comma from the skills.
  - **Excess whitespaces**.
  - **Non-ASCII characters**.
  - **Stop words**, like "the", "a", "an".
- **Standardized the vocabulary**, e.g. Data Analytics and Data Analysis were all replaced with "Data Analytics"
- **Lowercase**. We converted all capital letters in small letters.
- **Lemmatization**. Lemmatization, is the process of grouping together the inflected forms of a word so they can be analysed as a single item, identified by the word's lemma, or dictionary form. Unlike stemming, lemmatisation depends on correctly identifying the intended part of speech and meaning of a word in a sentence, as

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

well as within the larger context surrounding that sentence, such as neighboring sentences or even an entire document[Wik20c]. In an unstructured text, the same word may represent two meanings–for example, the word "lie" could mean to lay down and to tell something untruthful when used as a noun. While lemmatization would try to distinguish these two word senses, stemming would incorrectly conflate them.

3.2.3 Recommendation Engine

Architecture

Based on the user's query-interest, our system uses content-based filtering to generate a user profile, which is then used to recommend complementary courses for a job specialization. It combines gained skills from courses with the required skills from user's desired job specialization. In our approach, the features to represent both courses and jobs for the content-based algorithm were skills.

Figure 3.3 shows how our recommendation engine works with cosine similarity.

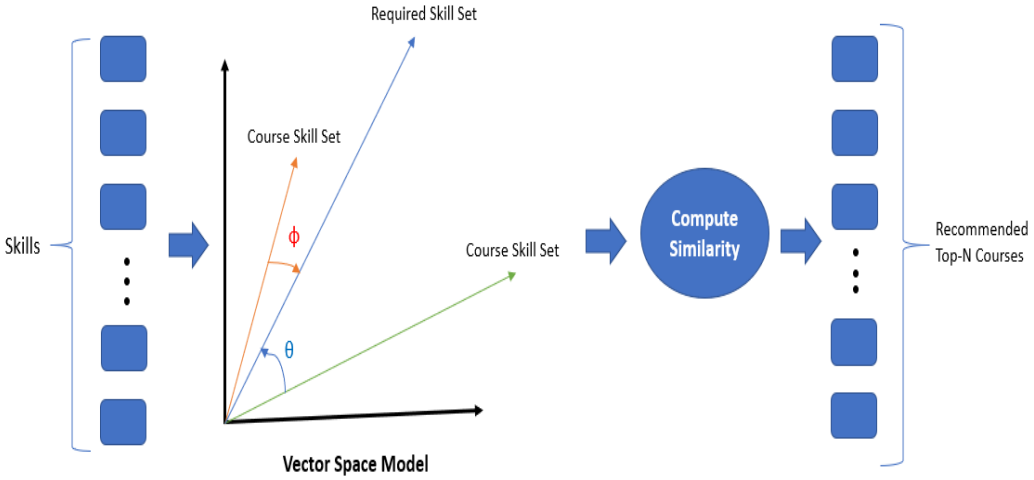


Fig. 3.3: Recommendation Engine

Vector Space Model

Vector Space Model was used to represent both required skill set for the desired job specialization, and skills sets, gained from courses. We experimented with two methods: *TF-IDF Vectorizer* and *CountVectorizer*.

TF-IDF: short for term frequency-inverse document frequency, is a text mining technique to weigh a term according to its importance in a document: the higher the term frequency,

the larger its weight will be. At the same time, it weighs the term inverse to its frequency across all documents. That means words such as the, a, and is which are likely to show up in multiple documents but are not useful for recommendation are weighed less than words that are more unique to the document. TF-IDF can be computed for a term using the formula shown in Equation 3.1.

$$W_{t,d} = \begin{cases} 1 + \log_{10} tf_{t,d}, & \text{if } tf_{t,d} > 0 \\ 0, & \text{otherwise} \end{cases} \tag{3.1}$$

Count Vectorizer(One-Hot Encoding): Is a very simple technique to convert text documents to vectors. We create vectors that have a dimensionality equal to the size of our vocabulary, and if the text data features that vocab word, we put a 1 in that dimension. Every time we encounter that word again, we will increase the count, leaving 0s everywhere we did not find the word even once.

In our case, the two techniques*TF-IDF* and *CountVectorizer*, led as to the same results. This is due to the fact that *Zippia* is an organized job hunting site and the skills are fixed. So, we had the direct and correct skills required, without trying to extract useful data from a job description text.

Similarity Calculation

Once the features were represented as vectors, according to the Vector Space Model, we used *cosine similarity*. Cosine similarity is a metric used to measure how similar the documents are irrespective of their size. Mathematically, it measures the cosine of the angle between two vectors projected in a multi-dimensional space. The smaller the angle, the higher the cosine similarity.

We, also, used *Jaccard similarity* as another metric of the similarity between features. We calculated the intersection over union of the two skill-sets for this system. Basically, the more matched keywords and fewer unmatched keywords, the higher the similarity. The value of both similarity scores ranges from 0 to 1, with their value increasing with increasing similarity.

In our system, the two aforementioned techniques, led us again to the same results.

3.3 Design

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

3.3.1 Technical Stack

The recommendation system was built in Python and the following libraries and frameworks were used in the development process:

- **Web Scraping:** bs4, requests, BeautifulSoup
- **Keyword Extraction:** numpy, pandas
- **Text Processing:** nltk, re
- **Similarity calculation:** Scikit-Learn

3.3.2 Algorithm

The algorithm [1] that we used was straightforward and performed the following steps:

1. First, extract features out of the job specializations to create an item representation.
2. Second, define a similarity function for these item representations and courses.
3. Third, compute the similarity between items.
4. Finally, select *top-K* recommendations to recommend.

Algorithm 1 Content-Based Recommendation Algorithm

Input: Number of items to be recommended $K \in \mathcal{N}$, List of all items $Items$
Output: K items to be recommended

for each: $item \in Items$ **do**
1: Extract features from $items$
2: Compute the $cosine$ or $Jaccard\ similarity$ between items
3: Sort items with descending order
4: Select $top - K$ recommendations

return $top(K, items)$

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

4 Evaluation

Before setting out to evaluate our recommender system it is important to consider our datasets. The courses recommended by our system, were recommended due to the similarity scores of the skills that the user will gain form the courses and those that the job specialization require. So, one factor to generate our recommendations was that the similarities had to be high. Finally, to evaluate the quality and the overall effectiveness of our online course recommendations we used traditional measures of information retrieval evaluation: *Precision*, *Recall* and *F-score*.

4.1 Data Selection and Visualization

We constructed a word cloud of the most demanding skills for job specializations in Figure 4.1 and, also, a word cloud of the most popular gained skills from online courses in Figure 4.2.



Fig. 4.1: Skills Word Cloud of Job Specializations

In addition, we plotted a bar chart in Figure 4.3, that shows top 30 data skills required by most employers according to the job specializations that we selected and a bar chart in Figure 4.4, that shows the 30 most popular skills gained from online courses.

From the bar charts, it should be noted that more than 40% jobs require Java, 38% require Python, and around 30% require R. But according to the courses that *Coursera* offers, Python is the most popular skill that you gain in more than 70% of the courses, while java



Fig. 4.2: Skills Word Cloud of Online Courses

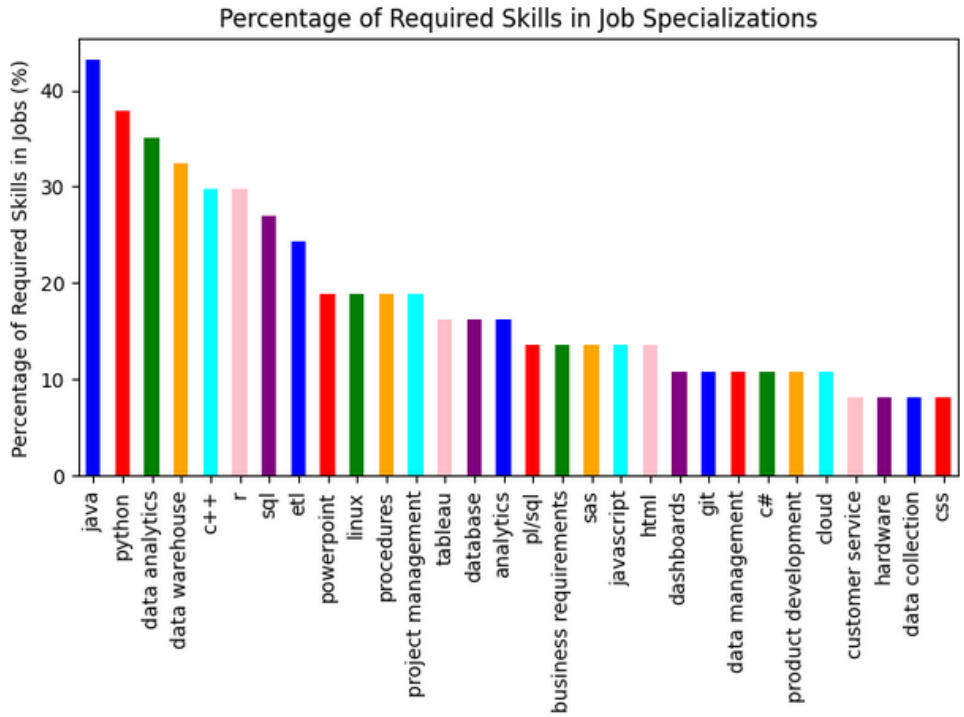


Fig. 4.3: Skills Required in Job Specializations

is not even in the top 30 skills! This results in recommending more courses with the most popular skills, and less courses with the least popular skills, like Java.

4.2 Evaluation Metrics

To test the quality of our recommendations, we evaluated recommendations generated by our system on the basis of *precision, recall* and *F-Score*.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

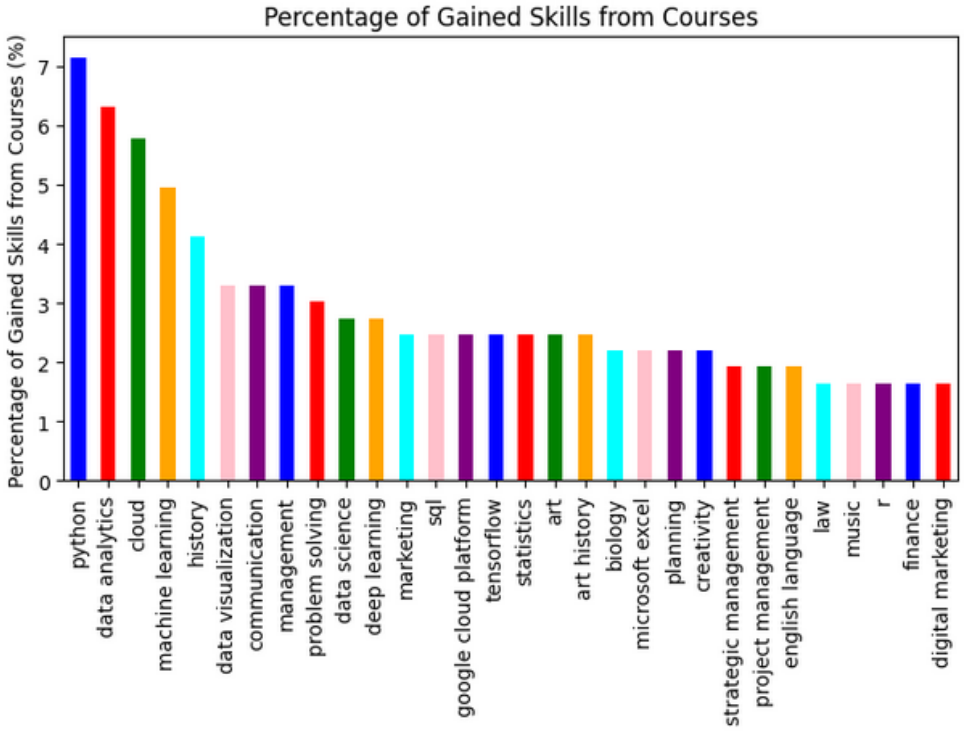


Fig. 4.4: Skills Gained from Online Courses

Precision is the fraction of relevant documents among the documents retrieved by the system. As an effectiveness measure, precision reflects how capable of delivering the relevant documents with the least amount of recommendation. It can be calculated using the formula 4.1:

$$Precision = \frac{|Recommendations \cap Relevant\ Documents|}{|Recommendations|} \tag{4.1}$$

Recall is the fraction of relevant documents contained in the set. Recall quantifies how capable is the model of finding all the relevant documents and recommend them to the user. It can be calculated using the formula 4.2:

$$Recall = \frac{|Recommendations \cap Relevant\ Documents|}{|Relevant\ Documents|} \tag{4.2}$$

F-score is the harmonic mean of precision and recall. It can be calculated using the formula 4.3:

$$F - score = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{4.3}$$

4.3 Experiment

In order to setup our experiment we executed the following queries per method:

- we chose 5 different job specializations from the field of data science, and
- for each specialization we considered two assumptions:
    1. we did not define the difficulty level of the user,
    2. we defined the difficulty level for "Beginners".

Our experiments follows the assumption that given a user's desired career path and the difficulty level, represented as a search query, the course recommendations returned are either relevant or irrelevant results. In this case, a course was considered "relevant" if it contained the one or more specific skills, or a generalization of the skill in its course title. For example, if the user queries for "Data Scientist" then a course would be considered relevant if it contained terms such as *"Python", "R", "Data Science", "Big Data", "Hadoop", "Java", "Tableau", "Data Visualization", "Tensorflow"* or *"Machine Learning"* in its title or skills.

The recommendations of an experiment for different similarity measures but with the same user's input specialization for beginners, are shown in Figures 4.5 and 4.6.

```
The 10 most recommended courses to Data Engineer are:

1 " Foundations for Big Data Analytics with SQL " ,
 with similarity score: 0.7970811413304554
2 " Modern Big Data Analysis with SQL " ,
 with similarity score: 0.7242243488319184
3 " Introduction to Data Analytics for Business " ,
 with similarity score: 0.6999132392733555
4 " Data-driven Decision Making " ,
 with similarity score: 0.6847623832623257
5 " Capstone: Retrieving, Processing, and Visualizing Data with Python " ,
 with similarity score: 0.6488856845230501
6 " Understanding and Visualizing Data with Python " ,
 with similarity score: 0.6488856845230501
7 " SQL for Data Science " ,
 with similarity score: 0.6488856845230501
8 " Essential Design Principles for Tableau " ,
 with similarity score: 0.6362847629757777
9 " Survey Data Collection and Analytics " ,
 with similarity score: 0.6295115802917068
10 " Digital Marketing Analytics in Theory " ,
 with similarity score: 0.6120563724821229
```

Fig. 4.5: The Top-10 Recommended Courses to User's Input, with Cosine Similarity

4.3.1 Experimental Setup

The two recommendation engines produced different similarity scores and courses' recommendations. In particular, cosine similarity's scores were higher and the reason is

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

```
The 10 most recommended courses to Data Engineer are:

1 " Foundations for Big Data Analytics with SQL " ,
 with similarity score: 0.36363636363636365
2 " Modern Big Data Analysis with SQL " ,
 with similarity score: 0.3125
3 " Databases and SQL for Data Science " ,
 with similarity score: 0.25
4 " Introduction to Data Science " ,
 with similarity score: 0.23076923076923078
5 " IBM Data Science " ,
 with similarity score: 0.2222222222222222
6 " Data-driven Decision Making " ,
 with similarity score: 0.16666666666666666
7 " SQL for Data Science " ,
 with similarity score: 0.16666666666666666
8 " Exploring and Preparing your Data with BigQuery " ,
 with similarity score: 0.16666666666666666
9 " Understanding and Visualizing Data with Python " ,
 with similarity score: 0.16666666666666666
10 " Python for Data Science and AI " ,
 with similarity score: 0.15384615384615385
```

Fig. 4.6: The Top-10 Recommended Courses to User's Input, with Jaccard Similarity

because Jaccard Similarity takes set of unique words, while cosine similarity increases with duplicate data, thus leading to different recommendations.

Moreover, some of the courses that cosine similarity provided us were irrelevant, while obtained high similarity score. This was due to the fact, that cosine similarity function took into account each word of the skills as a unique one, and not as a part of a skill set. For example, the skill "Data Analytics", was counted as two separate words. So, when the skill set contained a common word like "Data", the engine recommended not so relevant courses.

So, we changed the cosine function to consider the whole skill set as a unique word, even if it contained two or more words. This led us to the same recommendations for both similarity metrics, but with different similarity scores (Figure 4.7).

4.4 Evaluation approach

In this section we present the performance evaluation of the proposed system. In order to calculate the accuracy of the course recommendation algorithm we performed the following experiment.

- We selected 25 jobs specializations from the field of Computer Science. E.g., *Data Scientist, Data Engineer, Database Developer, Data Analyst, Business Analyst, Cloud Architect, Cloud Engineer, Platform Engineer, Database Analyst, Database Administrator, BI Engineer, BI Developer, BI Analyst, ML Engineer, AI Specialist, Web Analyst.*

```
The 10 most recommended courses to Data Engineer are:

1 " Foundations for Big Data Analytics with SQL " ,
 with similarity score: 0.5656854249492379
2 " Modern Big Data Analysis with SQL " ,
 with similarity score: 0.4767312946227962
3 " Databases and SQL for Data Science " ,
 with similarity score: 0.42426406871192845
4 " Introduction to Data Science " ,
 with similarity score: 0.3872983346207417
5 " IBM Data Science " ,
 with similarity score: 0.3651483716701107
6 " Data-driven Decision Making " ,
 with similarity score: 0.31622776601683794
7 " SQL for Data Science " ,
 with similarity score: 0.31622776601683794
8 " Exploring and Preparing your Data with BigQuery " ,
 with similarity score: 0.31622776601683794
9 " Understanding and Visualizing Data with Python " ,
 with similarity score: 0.31622776601683794
10 " Python for Data Science and AI " ,
 with similarity score: 0.28284271247461895
```

Fig. 4.7: The Top-10 Recommended Courses to User's Input, with Cosine Similarity

- For every specialization we made two assumptions. First, that the user has none of the skills that these specializations require. And, second, that the user posses one to three random skills, and we removed these skills from the required skill set of each specialization.
- For each of these specializations we vary the parameter of difficulty and calculate the affect that causes in our system. In particular we used 3 difficulty levels: none, Beginners and Advanced. For each specialization and for every of the above difficulty levels, our system recommended online courses, supposing that these courses are "relevant".
- We then calculated the number of all possible relevant items. To achieve this we used two different datasets from *Coursera*:

    1. **Job-based dataset:** We queried in *Coursera.org* each of the 25 job specializations with a difficulty level respectively, and crawled the courses,
    2. **Skills-based dataset:** We queried in *Coursera.org* every relevant skill that each job specialization requires with a difficulty level respectively, and crawled the courses,

- Finally, we computed the precision, recall and F-scores for each query.

4.5 Results

Based on the results of our experiment with both different datasets of relevant courses from *Coursera*, it would appear that:

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

- our algorithm performed better when we used the skills-based dataset from *Coursera*. In particular:
  - the recall was on average very close to 1. That means that most of the courses that our system recommended was considered relevant, because they were the courses that *Coursera* also recommended.
  - the precision and F-Score were lower than expected. This is because our system recommended courses that it considered "relevant", but *Coursera* did not recommend.

- our system performed not as good as we expected, particularly in the category of precision and F-score, when we used the job-based dataset form *Coursera*. This is due to the fact that:
  - *Coursera* depends to a large extent on whether the query-job specialization is included in the course title in the contrary with our algorithm which gives the same weight in title and skills.
  - *zippia*, which is the site that we scrapped the required skills for job specializations, has defined specific skills for each job specialization, based on certain criteria. These skills, though, may vary and depend from the country. So, *Coursera* may considers that the job specializations have different skills.
  - The recommendations made with both similarity measures *Cosine Similarity* and *Jaccard Similarity* essentially take for input user's job specialization, extract the required skills, transform the query into a new query based on these skills, and then compute the similarity score between the skill and course. Due to the transformations of the original requests, the recommendations of these algorithms and of queries based strictly on context will be slightly different.
  - *Coursera* provides more courses when the difficulty level is not defined and for Beginners. As a result, the algorithm performed better on average for these two levels than with advanced level, and the mean metrics became lower.

Table 4.1 and Figure 4.8 shows the different metrics: mean Precision, mean Recall and mean F-Score of our experiment, for both datasets of *Coursera* for relevant courses: job-based and skills-based, when the level is not defined.

Recommendation Algorithm	Evaluation Method	Dataset	
		Job-based	Skills-based
Content-Based Algorithm	Mean Precision	0.57	0.78
	Mean Recall	0.69	0.92
	Mean F-Score	0.6	0.81

Tab. 4.1: Experiment Evaluation Table

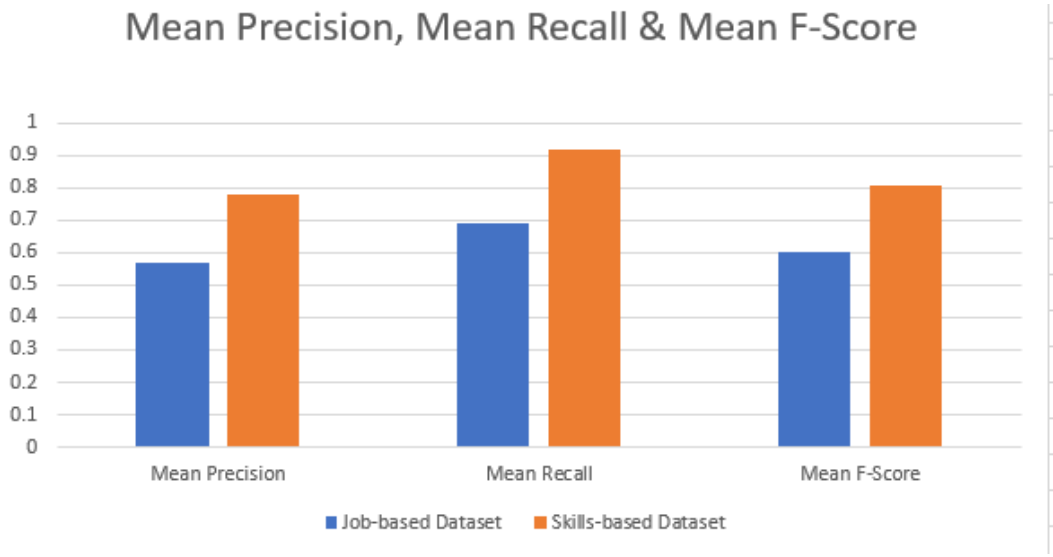


Fig. 4.8: Experiment Evaluation Bar Chart

4.5.1 Experiment Case Studies

The following are some case studies of our experiment for better understanding of the evaluation as a whole, and for comparing the results of the two above mentioned datasets for relevant courses from *Coursera*. In each of these case studies we used a specific job specialization, Data Scientist, the similarity metric was cosine similarity and the difficulty level varied from beginners to advanced and no defined level at all.

Case Study with No Difficulty Level:

Our system executed the following steps:

1. User queried Job A, which in our case is "Data Scientist", in the algorithm and the system extracted the required skills.
2. Job A requires the skills: *Python, R, Data Science, Big Data, Hadoop, Java, Tableau, Data Visualization, Tensorflow, Machine Learning.*
3. We defined the similarity score to be higher than 0. This means that every recommended course must contain at least one required skill or the term of Job A must be in the course title.
4. Our system recommended 121 courses. Among these courses are: *"Python Data Products for Predictive Analytics" -similarity score: 0.40, "Introduction to TensorFlow for Artificial Intelligence, Machine Learning, and Deep Learning"- similarity score: 0.37, "IBM AI Engineering"-similarity score: 0.32, "Machine Learning Foundations: A Case Study Approach"-similarity score: 0.32, "Introduction to Big Data"-similarity score: 0.32, "Essential Design Principles for Tableau"-similarity score: 0.32, "Applied Machine Learning in Python"-similarity score: 0.32, "Applied Plotting, Charting Data*

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Representation in Python"-similarity score: 0.32, "Capstone: Retrieving, Processing, and Visualizing Data with Python"-similarity score: 0.32, "Data-driven Decision Making"-similarity score: 0.32.

5. The similarity score ranged from *0.42-four matched skills per course, to 0.09-one matched skill.* The more matched skills, the higher the similarity score. And the total similarity score were up to 100%, which means that the recommended courses covered all the required skills.
6. Some of the gained skills are: *Machine Learning, Python, Predictive Analytics, Data Processing, Data Visualization, Python Libraries, Data Pre-Processing, Web Scraping, Big Data, Recommender Systems, Computer Vision, Tensorflow, Machine Learning, Data Science, Artificial Neural Network, Artificial Intelligence (AI), Machine Learning, Spark, Deep Learning, keras, Hadoop, Mapreduce, Cloudera, Data Analytics, Tableau, Data Virtualization, Data Visualization, Machine Learning (ML) Algorithms, Scikit-Learn, Matplotlib, Database , Data-Informed Decision-Making , Data Analytics.*
7. In this step we retrieved the relevant courses from *Coursera* with Job-based search and with Skills-based search:
   - With Job-based search we retrieved 87 relevant courses. Some of these courses are: *"IBM Data Science", "Introduction to Data Science", "Introduction to Big Data", "Excel to MySQL: Analytic Techniques for Business Specialization", "Genomic Data Science", "The Data Scientist's Toolbox", "A Crash Course in Data Science", "Machine Learning Foundations: A Case Study Approach ", "Supervised Learning: Regression", "AI and Machine Learning MasterTrack™ Certificate".*
   - With Skills-based search we retrieved 110 relevant courses. Some of these courses are: *"IBM Data Science", "Introduction to Data Science", "Introduction to Big Data", "Deep Learning", "IBM Data Analyst", "A Crash Course in Data Science", "Machine Learning Foundations: A Case Study Approach ", "Python for Everybody", "Applied Data Science".*
8. Finally, we estimated *precision, recall* and *F-Score* with both datasets and compared the results.

The results of our experiment for both datasets are shown in Table 4.2 and in Figure 4.9.

Recommendation Algorithm	Evaluation Method	Dataset	
		Job-based Search	Skills-based Search
Content-Based Algorithm	Precision	0.44	0.81
	Recall	0.48	0.89
	F-score	0.46	0.85

Tab. 4.2: Experiment Evaluation Table

It must be noted that from the first top 10 recommended courses, the similarity score of our system reached 80%. So, it provided courses that covered the most required skills, with

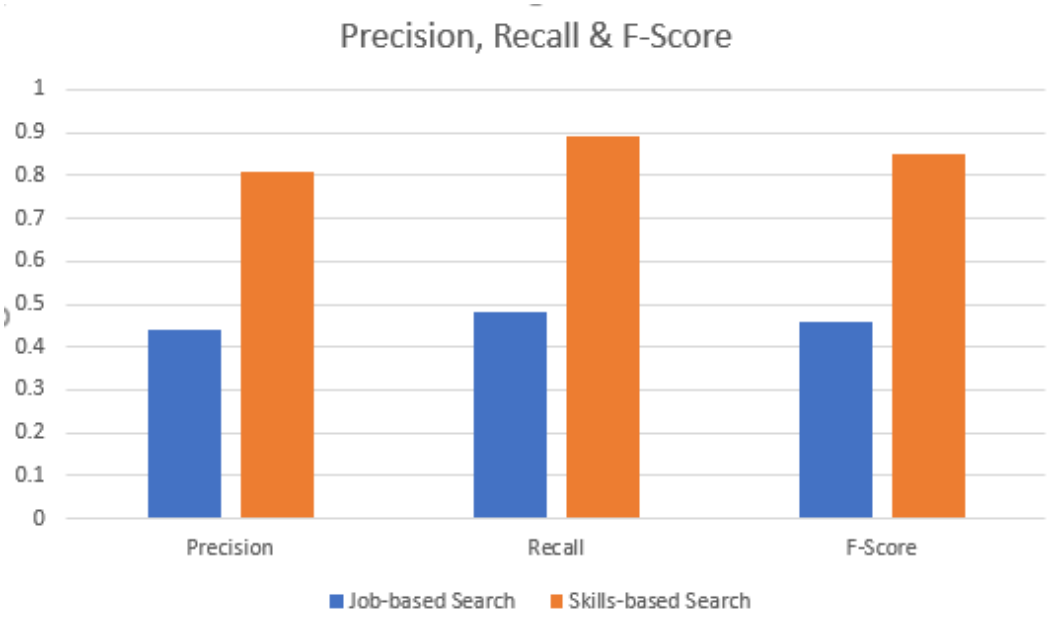


Fig. 4.9: Experiment Evaluation Bar Chart

Java and R being the only skills that our algorithm did not cover. This is due to the fact, that *Coursera* does not offer many courses with these particular skills.

Case Study with Difficulty Level for Beginners:

Our system executed the following steps:

1. User queried Job A, "Data Scientist", in our algorithm and difficulty level for Beginners. The system extracted the required skills.
2. Job A requires the skills: *Python, R, Data Science, Big Data, Hadoop, Java, Tableau, Data Visualization, Tensorflow, Machine Learning.*
3. We defined the similarity score to be higher than 0. This means that every recommended course must contain at least one required skill or the term of Job A must be in the course title.
4. Our system recommended 60 courses. Among these courses are: *"Applied Data Science" -similarity score: 0.38, "Data-driven Decision Making"- similarity score: 0.32, "Applied Data Science"-similarity score: 0.32, "Understanding and Visualizing Data with Python"-similarity score: 0.32, "Fundamentals of Visualization with Tableau"-similarity score: 0.32, "Introduction to Artificial Intelligence (AI)"-similarity score: 0.29, "Python for Data Science and AI"-similarity score: 0.29, "Data Analytics with Python"-similarity score: 0.29, "IBM Applied AI"-similarity score: 0.27, "IBM Data Science"-similarity score: 0.27.*
5. The similarity score was lower than with no level defined and ranged from *0.38-four matched skills per course, to 0.1-one matched skill.* The total similarity score were up to 100%, which means that the recommended courses covered all the required skills.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

6. Some of the gained skills are: *Data Science, Machine Learning, Python, Data Analytics, Data Visualization, Pandas, Numpy, Predictive Modelling, Model Selection, Data Virtualization, Matplotlib, Data-Informed Decision-Making, Big Data, Hadoop, Tableau, Statistics, Visualization (Computer Graphics), Deep Learning, Artificial Intelligence (AI), Jobs, Watson (Computer), API, Statistical Analysis, Business Intelligence, Cloud, Ipython, Relational Database Management System (RDBMS), SQL.*
7. In this step we retrieved the relevant courses from *Coursera* with Job-based search and with Skills-based search:
    - With Job-based search and difficulty level for beginners we retrieved 49 relevant courses. Some of these courses are: *"IBM Data Science", "Introduction to Data Science", "Python for Data Science and AI", "Excel to MySQL: Analytic Techniques for Business", "Data Science Fundamentals with Python and SQL", "Applied Data Science", "Statistics with R", "Big Data", "Introduction to Data Analytics", "SQL for Data Science"*
    - With Skills-based search and difficulty level for beginners we retrieved 55 relevant courses. Some of these courses are: *"IBM Data Science", "Introduction to Data Science", "Introduction to Big Data", "Deep Learning", "IBM Data Analyst", "A Crash Course in Data Science", "Machine Learning Foundations: A Case Study Approach ", "Python for Everybody", "Applied Data Science"*.
8. Finally, we estimated *precision, recall* and *F-Score* with both datasets and compared the results.

The results of our experiment for both similarity measures are shown in Table 4.3 and in Figure 4.10.

Recommendation Algorithm	Evaluation Method	Datasets	
		Job-based Search	Skills-based Search
Content-Based Algorithm	Precision	0.53	0.83
	Recall	0.65	0.90
	F-score	0.58	0.86

Tab. 4.3: Experiment Evaluation Table

Similarly in this experiment, from the first top 10 recommended courses, the similarity score of our system reached 80%. So, it provided courses that covered the most required skills, with *Java* and *Tensorflow* being the only skills that our algorithm did not cover. In addition, in this experiment the evaluation scores were better, mainly because the courses were less.

Case Study with Advanced Difficulty Level:

Our system executed the following steps:

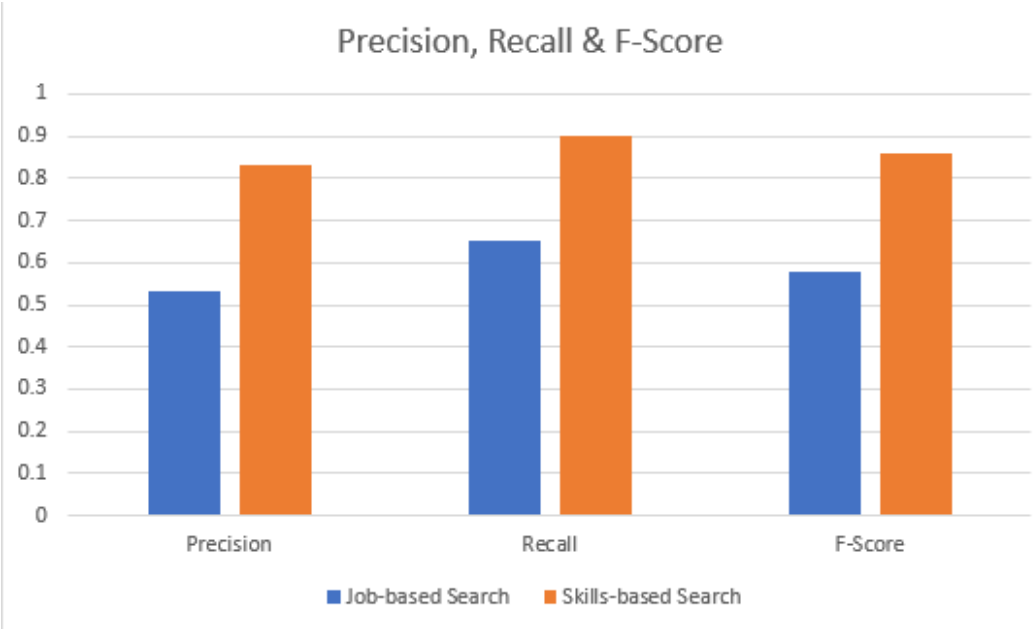


Fig. 4.10: Experiment Evaluation Bar Chart

1. User queried Job A, "Data Scientist", in our algorithm and difficulty level "Advanced". The system extracted the required skills.
2. Job A requires the skills: *Python, R, Data Science, Big Data, Hadoop, Java, Tableau, Data Visualization, Tensorflow, Machine Learning.*
3. We defined the similarity score to be higher than 0. This means that every recommended course must contain at least one required skill or the term of Job A must be contained in the course title.
4. Our system recommended only 6 courses: *"IBM AI Enterprise Workflow" -similarity score: 0.42, "Advanced Data Science with IBM"- similarity score: 0.24, "Advanced Machine Learning with TensorFlow on Google Cloud Platform"-similarity score: 0.24, "Advanced Statistics for Data Science"-similarity score: 0.12, " Strategic Business Analytics"-similarity score: 0.11, "Data Warehousing for Business Intelligence"-similarity score: 0.09.*
5. The similarity score ranged from *0.42-three matched skills, to 0.09-one matched skill.* The total similarity score were up to 60%, which means that the recommended courses did not cover all the required skills.
6. The total gained skills are: *Data Science, Information Engineering, Artificial Intelligence (AI), Machine Learning, Python, Internet Of Things (IOT), Deep Learning, Spark, Statistics, Long Short-Term Memory (ISTM), Tensorflow, Convolutional Neural Network, Estimator, Advanced Machine Learning, R, Linear Regression, Statistical Hypothesis Testing, Statistics, Linear Algebra, Confidence Interval, Biostatistics, Marketing Analytics, Presentation, Marketing Performance Measurement And Management, Data Analytics, Business Analytics, Market Segmentation, Customer Lifetime Value, Pentaho, Data Visualization, Data Warehouse, SQL, Database (DB) Design, Entity–Relationship (E-R) Model, Database, Extraction, Transformation And Loading (ETL), Data Integration,*

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Materialized View, Business Intelligence. So, the required skills that the recommended courses didn't provide, were *Java, Big Data, Hadoop* and *Tableau.*

7. In this step we retrieved the relevant courses from *Coursera* with Job-based search and with Skills-based search:

    - With Job-based search and advanced difficulty level we retrieved 8 relevant courses: *"IBM AI Enterprise Workflow", "Advanced Data Science with IBM", "Applied AI with Deep Learning", "AI Workflow: Data Analysis and Hypothesis Testing", "AI Workflow: Feature Engineering and Bias Detection", "Advanced Machine Learning with TensorFlow on Google Cloud Platform", "AI Workflow: Machine Learning, Visual Recognition and NLP", "AI Workflow: Enterprise Model Deployment".*
    - With Skills-based search and advanced difficulty level we retrieved 5 relevant courses: *"Advanced Machine Learning with TensorFlow on Google Cloud", "Advanced Data Science with IBM, "IBM Enterprise Workflow", "Strategic Business Analytics", "Natural Language Processing".*

8. Finally, we estimated *precision, recall* and *F-Score* with both datasets and compared the results.

The results of our experiment for both similarity measures are shown in Table 4.3 and in Figure 4.11 .

Recommendation Algorithm	Evaluation Method	Datasets	
		Job-based Search	Skills-based Search
Content-Based Algorithm	Precision	0.5	0.67
	Recall	0.38	0.80
	F-score	0.43	0.73

Tab. 4.4: Experiment Evaluation Table

Case Study with Difficulty Level for Beginners, where the user posses 3 of 10 required skills:

Our system executed the following steps:

1. User queried Job A, "Data Scientist", in our algorithm and difficulty level for Beginners. The system extracted the required skills.
2. Job A requires the skills: *Python, R, Data Science, Big Data, Hadoop, Java, Tableau, Data Visualization, Tensorflow, Machine Learning.*
3. The user removed from the above skill set: *Python, Machine Learning, Tensorflow,* because she has the specific skills.



Fig. 4.11: Experiment Evaluation Bar Chart

4. We defined the similarity score to be higher than 0. This means that every recommended course must contain at least one required skill or the term of Job A must be in the course title.
5. Our system recommended 47 courses. Among these courses are: *"Data-driven Decision Making"- similarity score: 0.38, "Fundamentals of Visualization with Tableau"-similarity score: 0.38, "Applied Data Science"-similarity score: 0.23, "Information Visualization"-similarity score: 0.22, "Mastering Software Development in R"-similarity score: 0.22, "Data Science: Foundations using R"-similarity score: 0.22, "Big Data"-similarity score: 0.22.*
6. The similarity score ranged from *0.38-two matched skills per course, to 0.1-one matched skill.* The total similarity score were up to 100%, which means that the recommended courses covered all the required skills.
7. Some of the gained skills are: *Data-Informed Decision-Making , Big Data , Data Analytics, Tableau, Data Virtualization, Data Science , Machine Learning , Python, Data Visualization , Pandas , Numpy , Predictive Modelling , Model Selection, Matplotlib, Information Visualization (INFOVIS) , D3.Js, R , Ggplot2 , tidyverse , Object-Oriented Programming (OOP) , Data Manipulation , Regular Expression (REGEX) , Rstudio , Logic Programming , Functional Programming , Programming Tool, Github, Data Cleansing , Cluster Analysis, Neo4j , Mongodb , Spark , Hadoop , Mapreduce , Cloudera , Data Modeling , Data Management , Splunk.*
8. In this step we retrieved the relevant courses from *Coursera* with Job-based search and with Skills-based search:
    - With Job-based search and difficulty level for beginners we retrieved 49 relevant courses. Some of these courses are: *"IBM Data Science", "Introduction to Data Science", "Python for Data Science and AI", "Excel to MySQL: Analytic Techniques*

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

for Business", "Data Science Fundamentals with Python and SQL", "Applied Data Science", "Statistics with R", "Big Data", "Introduction to Data Analytics", "SQL for Data Science"

- With Skills-based search and difficulty level for beginners we retrieved 40 relevant courses. Some of these courses are: *"IBM Data Science", "Introduction to Data Science", "Introduction to Big Data", "Deep Learning", "IBM Data Analyst", "A Crash Course in Data Science", "Data Science: Foundations using R", "Java Programming and Software Engineering Fundamentals"*.

9. Finally, we estimated *precision, recall* and *F-Score* with both datasets and compared the results.

The results of our experiment for both similarity measures are shown in Table 4.5 and in Figure 4.12.

Recommendation Algorithm	Evaluation Method	Datasets	
		Job-based Search	Skills-based Search
Content-Based Algorithm	Precision	0.53	0.74
	Recall	0.51	0.88
	F-score	0.52	0.8

Tab. 4.5: Experiment Evaluation Table

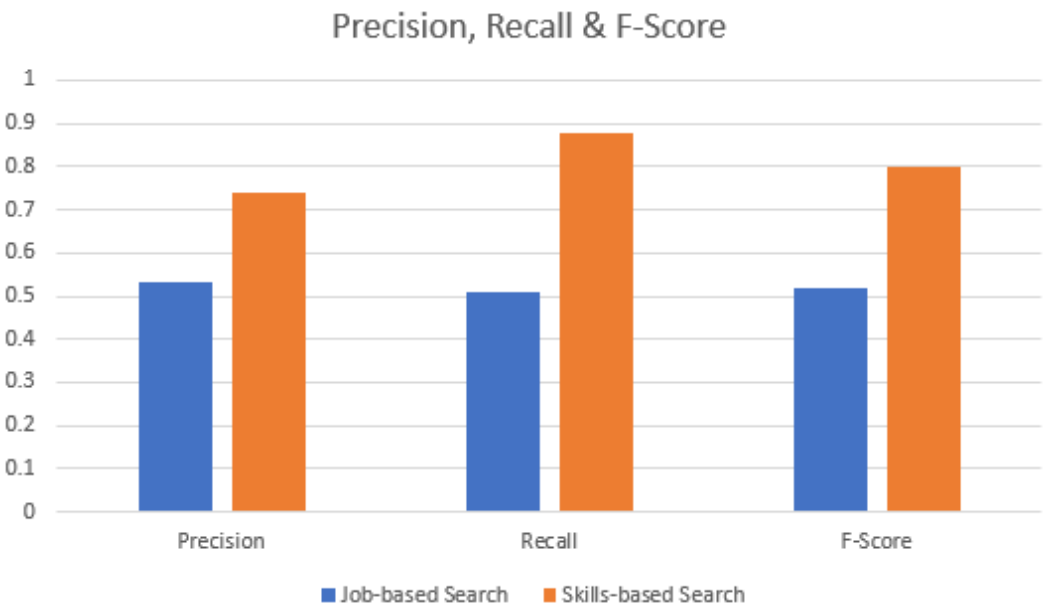


Fig. 4.12: Experiment Evaluation Bar Chart

In total, our system performed better with the skills-based dataset for relevant courses and the scores of our evaluation metrics were much higher.

One reason that our algorithm didn't perform as good as we expected when the relevant courses came up from *Coursera* by job-based search, is because this dataset is more focused on the course titles, that is whether the course title contains the job title, while our system is more focused on the skills. This has as an effect to low scores of the evaluation metrics, because many of our recommendations did not match with the relevant courses of *Coursera*.

Furthermore, our system performed just as well when we removed the skills that the user already possesed.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

5 Conclusions

In this paper, we presented an online course recommender system that uses content-based filtering to generate a user profile and suggests online courses and required skills to users based on their desired career specialization. We observed that users who use personalized online course recommendations to obtain new skills, are most likely to enroll to a more relevant course and to acquire the qualifications for a career on their interest faster than users who did not take advantage of personalized recommendations.

In the future, we could build a web application, where users could provide either the specialization they desire or directly a list of skills they are interested in learning and their resume. Furthermore, we could integrate more online courses' platforms like *edX*, *Udemy* and *LinkedIn*, in order to include a wide variety of courses and skills, we could extend the list of jobs' specializations and give the option to the user to choose either junior, mid or senior level. Finally, one drawback of our system is that it provides several courses with similar gained skills, so we could improve it by removing the courses with similar gained skills.

In conclusion, our approach offers several advantages over prior course recommender systems and turns a simple online course recommender system into a discovery tool for both relevant skills and courses. It empowers users to discover exactly what is expected from them to get the job of their dreams. In addition, it helps recruiters to choose between resumes that better cater to industry demands.


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Bibliography

[Apa+14] Rel Guzman Apaza, Elizabeth Veronica Vera Cervantes, L. Quispe, and José Eduardo Ochoa Luna. "Online Courses Recommendation based on LDA". In: *SIMBig*. 2014 (cit. on p. 11).

[Das15] Shuvayan Das. *Beginners Guide to learn about Content Based Recommender Engines*. 2015. URL: https://www.analyticsvidhya.com/blog/2015/08/beginners-guide-learn-content-based-recommender-systems/ (visited on Aug. 11, 2015) (cit. on p. 5).

[Est+17] David Estrela, Sérgio Batista, Diogo Martinho, and Goreti Marreiros. "A Recommendation System for Online Courses". In: *Recent Advances in Information Systems and Technologies*. Ed. by Álvaro Rocha, Ana Maria Correia, Hojjat Adeli, Luís Paulo Reis, and Sandra Costanzo. Cham: Springer International Publishing, 2017, pp. 195–204 (cit. on p. 12).

[KRK20] Pradnya Kulkarni, Sunil Rai, and Rohini Kale. "Recommender System in eLearning: A Survey". In: Jan. 2020, pp. 119–126 (cit. on p. 14).

[MGR17] Marouane El Mabrouk, Salma Gaou, and Mohamed Kamal Rtili. "Towards an Intelligent Hybrid Recommendation System for E-Learning Platforms Using Data Mining". In: *International Journal of Emerging Technologies in Learning (iJET)* 12.06 (2017), pp. 52–76 (cit. on p. 13).

[ODA19] R. Obeidat, R. Duwairi, and A. Al-Aiad. "A Collaborative Recommendation System for Online Courses Recommendations". In: *2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML)*. 2019, pp. 49–54 (cit. on p. 13).

[Pin17] Carlos Pinela. *Recommender Systems — User-Based and Item-Based Collaborative Filtering*. 2017. URL: https://medium.com/@cfpinela/recommender-systems-user-based-and-item-based-collaborative-filtering-5d5f375a127f (visited on Nov. 6, 2017) (cit. on p. 10).

[PKE17] B. Patel, V. Kakuste, and M. Eirinaki. "CaPaR: A Career Path Recommendation Framework". In: *2017 IEEE Third International Conference on Big Data Computing Service and Applications (BigDataService)*. Los Alamitos, CA, USA: IEEE Computer Society, Apr. 2017, pp. 23–30 (cit. on p. 13).

[Sha19] Shehba Shahab. "NEXT LEVEL: A COURSE RECOMMENDER SYSTEM BASED ON CAREER INTERESTS". PhD thesis. San Jose State University, 2019 (cit. on p. 10).

[SM16] P. Symeonidis and Dimitrios Malakoudis. "MoocRec.com : Massive Open Online Courses Recommender System". In: *RecSys Posters*. 2016 (cit. on p. 12).

[Spa20] Cambridge Spark. *Implemented your own recommender systems in Python*. 2020. URL: https://blog.cambridgespark.com/nowadays-recommender-systems-are-used-to-personalize-your-experience-on-the-web-telling-you-what-120f39b89c3c (visited on Jan. 23, 2020) (cit. on p. 10).

[Wan+17] H. Wang, C. Chootong, A. Ochirbat, et al. "Online courses recommendation system based on industry occupation skills requirements". In: *2017 10th International Conference on Ubi-media Computing and Workshops (Ubi-Media)*. 2017, pp. 1–6 (cit. on p. 12).

[Wik20a] Wikipedia. *Collaborative filtering*. 2020. URL: https://en.wikipedia.org/wiki/Collaborative_filtering (visited on Oct. 14, 2020) (cit. on p. 11).

[Wik20b] Wikipedia. *Jaccard index*. 2020. URL: https://en.wikipedia.org/wiki/Jaccard_index (visited on Sept. 6, 2020) (cit. on p. 7).

[Wik20c] Wikipedia. *Lemmatisation*. 2020. URL: https://en.wikipedia.org/wiki/Lemmatisation (visited on Oct. 12, 2020) (cit. on p. 19).

[Wik20d] Wikipedia. *Recommender system*. 2020. URL: https://en.wikipedia.org/wiki/Recommender_system (visited on Oct. 19, 2020) (cit. on p. 1).

[Wik20e] Wikipedia. *tf–idf*. 2020. URL: https://en.wikipedia.org/wiki/Tf-idf (visited on Oct. 11, 2020) (cit. on p. 6).

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

List of Acronyms

CF Collaborative Filtering

RS Recommender System

WWW World Wide Web


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

List of Figures

2.1 Vector Space Model 6
2.2 Jaccard similarity 7
2.3 Collaborative Filtering 8
2.4 Memory-based Techniques 9
2.5 User-Item Matrix based on k Neighbors Ratings 10

3.1 Architecture Diagram of the Recommendation System 16
3.2 Course Ratings' Distribution 18
3.3 Recommendation Engine 19

4.1 Skills Word Cloud of Job Specializations 23
4.2 Skills Word Cloud of Online Courses 24
4.3 Skills Required in Job Specializations 24
4.4 Skills Gained from Online Courses 25
4.5 The Top-10 Recommended Courses to User's Input, with Cosine Similarity . 26
4.6 The Top-10 Recommended Courses to User's Input, with Jaccard Similarity . 27
4.7 The Top-10 Recommended Courses to User's Input, with Cosine Similarity . 28
4.8 Experiment Evaluation Bar Chart 30
4.9 Experiment Evaluation Bar Chart 32
4.10 Experiment Evaluation Bar Chart 34
4.11 Experiment Evaluation Bar Chart 36
4.12 Experiment Evaluation Bar Chart 37

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

List of Tables

4.1 Experiment Evaluation Table 29
4.2 Experiment Evaluation Table 31
4.3 Experiment Evaluation Table 33
4.4 Experiment Evaluation Table 35
4.5 Experiment Evaluation Table 37

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

List of Algorithms

1 Content-Based Recommendation Algorithm 21


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ