Alkadhum Journal of Science, Vol. 1, No. 2 (2023)

Open Access ISSN: 3005-2386 (Online), ISSN: 3005-2378 (Print)

IRAQI Academic Scientific Journals

Alkadhum Journal of Science (AKJS)

Journal Homepage: https://alkadhum-col.edu.iq/JKCEAS

Alkadhum Journal of Science

An Explainable Content-Based Course Recommender Using Job Skills

Yasir M. Al-Sarraf*

Department of Information Technology, Imam Al-Kadhum College, Iraq

Article information

Article history:

Received: November, 5, 2023

Accepted: December, 5, 2023

Available online: December, 14, 2023

Keywords:,
Explainability,
Job requirement,
Recommendation system

Corresponding Author:
Yasir Mahmood Younus
alsarraf@iku.edu.iq

DOI:
https://doi.org/10.61710/akjs.v1i2.62

Abstract

The large number of courses offered in universities and online studies made it difficult for students to choose the courses that suit their interests and career goals, which led students to lose many opportunities to be employed in the job they wanted. To keep pace with the rapid development of technology, and instead of relying on the job title as was previously done, the employers began to identify the skills required for a job. The competencies of the candidates are then examined and evaluated according to those requirements. Thus, it has become necessary for students to take courses that suit their future professional interests, ensuring that they are employed in the job they desire and supporting their long-term career success. Fortunately, the emergence of skills-based employment has provided an opportunity for universities and colleges to create a clearer path to the courses offered to allow students to take courses that match their future career interests. In this study, we used K-Mean clustering algorithm, TF-idf approach, and content-based filtering algorithm to provide relevant courses for students based on the required job with an explanation of why these courses are recommended. Our result illustrates that our method offers many advantages compared with other recommender systems. our system converts a simple course recommendation into a tool for discovering skills. Since many recommendation systems work as black boxes, we designed our system to recommend the relevant course with explaining why these courses are recommended, which will add a factor of transparency to our system and confirms the reliability of the system to the students.

1. Introduction

Planning is the secret of success. Many young people today are thinking about their futures. And they put a plan that they imagine will lead them to achieve a stable life. Many young people believe that completing their high school studies and entering university will provide them with sufficient knowledge and skills to get the job that meets their future ambitions. Some of them know the path that will lead them to achieve their goals, Others do not have a clear vision of their path, which will lead them to not achieve their goals and maybe fail. Previously, the prevailing belief of many, is that the courses they study in universities will provide them with the skills required for employment. However, the major change in the needs of the labor market caused a lot of confusion to the graduated students as a result of their lack of ownership of the skills required for employment. Besides, today's important job opportunities may become irrelevant tomorrow due to technological progress. For this reason, a person must possess sufficient knowledge of the labor market and an accurate appreciation of its developments to make him more likely to succeed in his future life plan. With the arrival of the era of big data and the availability of many online learning resources, such as universities and colleges sharing information on their web platform, the challenge has become to know the content of these courses correctly and to know whether they will provide students the skills they need to acquire to achieve their career goals [1]. For this reason, course recommender systems are used to suggest courses to students based on what they require, such as (required job, High-Grade, Interest, etc.) [1][2]. In this study, by examining and processing the information related to the courses from the EDX website, and jobs from the SEEK website, we recommend courses for students that enable them to obtain the skills required for the job they desire.

2. Recommender Approach

Recommender systems can be categorized into three basic systems (Content-Based Filtering, Collaborative Filtering, Hybrid Recommenders).

2.1 Content-Based Filtering (CBF)

Content-based filtering concentrate on item characteristics. Similarities of items are determined by measuring the resemblance in their properties [3].

2.2 Collaborative Filtering (CF)

Collaborative filtering frameworks for (CF) concentrate on the user-item relationship. The resemblance of items is defined by the resemblance of the ratings of those items through users who have rated both items [4].

2.3 Hybrid Recommenders

Hybrid systems of recommendation merge the two basic approaches [5][6].

2.4 Explanations in recommender systems

Recommender Systems (RSs) These are new technologies that have been used recently due to the massive increase in the volume of data to help suggest items that are relevant to users' interests. These software techniques have been used in many fields such as research articles, books, social media, news, movies, search queries, etc. [7]. Recommendation systems recommend items to users based on their preferences and interests. During the past period, many machine learning algorithms were used to provide different recommendations to the users. In most cases, these recommendations were unclear, and the users did not understand them. In other words, they were black boxes and did not explain to the user any information about the reason for the recommendation. Which called for the need to provide recommendations accompanied by explanations for the reason for the recommendation. Therefore, adding this clarification to the recommendation will provide support to users and help them make decisions [8]. The explanation provided with the recommendation usually contributes to providing users with a broader understanding of whether the proposed component is suitable for their needs or not. Thus, the explanation added transparency to the recommendations and provided confidence for users to use the system. In the past few years, due to the great demand for the use of recommendation systems, it has become important to provide explanations automatically with recommendations. This is what Amazon started using for online trading. Explanations were represented in different forms, each according to its location and recipient, some textual and others directed using forms of graphs or diagrams. Thus, these recommendations quickly provided a broader understanding to the user about the reasons for the recommendations [9].

3. Related Work

Various approaches have been used in applications for course recommendation by learning from historical enrollment data. First, Content-based filtering methods suggest a student's course by analyzing the content of the course and clustering the student and course into groups to achieve similarity. Secondly, collaborative filtering methods suggest a student's course by observing the similarity of the student with the students' historical data in the system and predicting which course the student will be interested in. Third, Association rules based on frequent patterns are used to discover interesting relationships that are described by student selections for previous courses. Recently, other methods including sequence discovery and representation education have been used in this domain [9]. B. Behkamal et al. In 2019, they proposed a heuristic method to discover students' favorite courses based on features extracted from educational data. the initial phase contains three tasks. First, select the required features from the FUM dataset. Then, six measures are defined to select interesting courses from students' points of view, and in the last step, the most favorite courses are extracted using the proposed measures. they used the educational dataset of Ferdowsi University of Mashhad (FUM) in their work for experiments. FUM is currently one of the nation's top three universities and also the largest center of higher education in North- East of Iran. As a result, the FUM dataset contains comprehensive information on students of various degrees (associate, bachelor, master, and Ph.D.) in 537 different fields of study. they have selected a portion of the FUM dataset containing master and doctoral courses because they have more freedom of choice when it comes to picking courses without any limitation. The candidate features used in this work are extracted from three tables of Student information, Grades, and Courses. then they performed a cleaning of data and preprocessing tasks. Also, they analyzed the effects of six heuristically proposed measures to discover the favorite courses from students' point of view, the six measures are (the number of students who picked the course, the GPA of students in each course, the variety of students who selected the course, the number of students who dropped the courses throughout the semester, the number of times that a course is offered, the average of master and doctoral students who register in a course) Then to Detecting Favorite Courses, since their method was depending on a heuristic methodology, there was no evaluation metric or previously achieved results to compare evaluate their work with others. they calculated the mean of all measures for the top 10 results and used it as the label. In the other phase, they extract frequently jointly selected courses using Apriori and ECLAT algorithms. By comparing two sets of courses obtained from the first and second phases, curriculum scheduling can be done in a manner that favorite and frequently selected courses do not have any time conflict. To attain a better, analyze of the results [10]. In another work by Pardos et al. in 2020, they deal with the issue of the filtrate bubble in the context of a course recommendation system in production at a public university. their line is to be offered course results that are new or unexpected to students but still related to their interests, they used a dataset having anonymized student course enrollments. From the fall of 2008 to the fall of 2017 at UC Berkeley. The dataset contains ~~of~~ per-semester course enrollment records for For164,196 students (students and graduates alike), with a total of 4.8 million registrations. A course register record means that the student was still joined in the course at the finale of the semester. Students at this university, during this period, were allowed to drop courses up until close to the finale of the semester without penalty. They present their method depending on three competing models that are used to create their views. The initial model uses course2vec for learning course representations from registration sequences. The second is a variant on course2vec, which learns the representations of features clearly defined of an, ~~as~~ an example (instructor and the department) furthermore to the course representation. The awareness behind this methodology is that the course representation can have, conflated in it, the effect of the multiple instructors that have trained the course over time. They believe this "disintegration" will improve the fidelity of the representation of the course and act as a more precise representation of the current nature of the course. The last model is a standard vector bag of terms, designed strictly from its catalog definition for each course. Finally, they explore concatenating a course's course2vec and bag-of-words representation vector, they used 5 algorithms in their models to recommend different courses based on serendipity {BOW (div), Analogy (div), Equivalency (div), Equivalency (non-div), RNN (non-div)}. A study was conducted to evaluate the quality of recommendations drawn from our different course representations. Users rated each course from each recommendation algorithm along five dimensions of quality. Students were requested to rate course recommendations in terms of their (1) unexpectedness (2) successfulness interest in taking the course (3) novelty (4) diversity of the results (5) and (6) the specified commonality between the results. By contrasting these models. the findings of the user study, RNN 's suggestions demonstrate a tragic lack of recommendation that makes it difficult to achieve serendipity. besides students found simple bag-of-words based recommendations more serendipitous [11]. Another work by Esteban, in 2020, suggests a hybrid RS that merges the Collaborative Filtering method and Content-based Filtering method using multiple criteria related both to student and course information to recommend the most suitable courses to the students. A Genetic Algorithm has been developed to automatically find out the best RS configuration which contains both the most pertinent criteria and the configuration of the rest of the parameters. they proposed a methodology that has numerous steps. First, it addresses the description and processing of the used data. after that, they explained the proposed system (hybrid multi-criteria system). This system proposes courses to university students depending on several criteria related to both student and course information Finally, the planned optimization approach is defined that assigns a weighting for each criterion and automatically optimizes the rest of the RS parameters. This approach allows the importance of each criterion to be defined using a weighting scheme. Thus, the most relevant criteria are higher, while the fewer are lower. The method also finds the best configuration for parameters including similarity measures and neighborhood sizes for the proposed RS. The experimental research used a real dataset from the University of Córdoba (Spain) Computer Science Department. This includes information obtained from students over three academic years, including 2,500 entries from 95 students and 63 courses. Experimental findings indicate that considering several criteria provides better results, but it is necessary to study how each of them is relevant because not all factor is evenly significant. Besides, using a hybrid system that merges both CF and CBF will enhance the results achieved [12]. Mondal et al. 2020, they proposed a machine-learning technique to recommend suitable courses to learners based on their learning history and past performance. Their framework will work based on historical and survey data. first, they collect data and then clean and select the process for the data collected. third, normalization which includes the integration of data from heterogeneous sources. after this step they refer to the data set by D. A clustering algorithm will be applied to the data set D to create a party of similar learners. Once the clusters of learners are created a frequent pattern mining algorithm will be applied for each cluster. The system classifies the students based on historical data by finding out what was the background of students who scored higher grades in each course. Every time a new student enters ~~in~~ the system will be classified using the clusters and a set of courses will be recommended to the learner based on frequent pattern mining. Further, based on an online test the adaptability of the learner will be tested to the customized recommended courses according to the learner's needs. The framework will provide a personalized environment of study to each learner. They compared three systems and through the experiment, they found that the proposed system is acceptable, efficient, and beneficial for students [2]. In another work done by MA et al. in 2020, that targets to recommend suitable courses for learners and study how to design a personalized course recommendation in the university environments, a Hybrid course they proposed to recommendation framework that considers student interest, the timing and popularity of courses, and predicted performance of students, simultaneously. Experiments were conducted to confirm the efficiency of their offered approach, they compared their method with two group popularity approaches, and Random recommendation (Random). The two group popularity approaches include the department level (Grp-Pop-1), which recommends the common courses in the major, and the academic level (Grp-Pop-2), which recommends the most common courses on the major and the academic level of the student (“freshmen”, “sophomores”, “juniors”, and “seniors”). The results show that the suggested hybrid course recommendation approach performed well compared to other methods. Also, the model itself is flexible in the public sense that one can easily adjust or extend it by changing the recommendation formula and incorporating more information [1]. Another work done by Guoqing Zhu1 et al. in 2020, They used the course data along with the job data. In their study, they specifically used the data provided as a description of the outputs of the training courses with the job data that represented the requirements for getting the job. By making use of this data and based on the student's professional goals, the courses were recommended. Their method is a unique approach that allows them to provide career recommendations to students based on their profiles. In their research, they used different sources. The first source was educational data, and the second source was job advertisements. They combined these sources to represent the skills acquired from the courses and required for employment, the link between this data, in other words, it was the bridge between these two sources of data. This data was linked through the use of the infomap algorithm. Then they applied a random walk algorithm to generate suggestions based on the functional objectives. They used Indiana University data for the computing and engineering departments as course data that contained course selection data for students for four years. The data used contains the information of more than 7800 students distributed over five departments. For 16 semesters, the course data consisted of approximately 380 courses and more than 180,000 associated records. These data lacked information on the skills that students will acquire as a result of their enrollment in these courses. Therefore, to initiate the implementation of the study, sufficient information must be provided about the skills provided by the courses to the students, so they used the greedy matching algorithm to obtain the skills relevant to the courses offered. Over 1,000 skills were extracted by utilizing the massive open online course MOOCs. These skills were related to several fields, including computer science and systems engineering, in addition to informatics and statistics. Of the 1,000 skills found, 367 were relevant to only 266 courses from Indiana University data. In the end, the course data consisted of four features, one of these features was the skills provided by the course. Regarding jobs data, job advertisements for the year 2019 were used from the Careerbuilder2 site. After processing this data, it included approximately 20,000 job advertisements related to approximately 1,600 skills. The jobs data included five attributes, including the name of the job, and another is the requirements for getting this job. They applied their studies according to three scenarios to recommend courses:

- Scenario 1: A university student is looking for courses that will provide him with the skills needed to get the job he wants.
- Scenario 2: A university student with some skills gained from the courses he attended previously needs to acquire other skills to achieve his career goal.
- Scenario 3: A junior employee or an employee with good experience wants to get a higher job by acquiring new skills.

In their experiments, they used two methods, one of these methods was using the vector space model and the other was using the probabilistic model. According to the results, their method outperformed the basic methods of most evaluation scales, except for the accuracy measure, the results were not satisfactory. Based on the results of the three scenarios, the recommendations were good for the professional students. One of the challenges was the use of data from two different sources, so the overlapping skills were few. They were able to assign less than 80 skills out of more than 370 skills that were predetermined. Based on what is stated in the conclusion, they plan to improve the quality of the output through the use of more comprehensive data, adding many features to both job data and course data, and using an advanced algorithm to improve the results of the system recommendations [13].

4. Research Approach

Figure (1) shows the general structure of the proposed system. As can be seen from the figure, the work can be divided into three stages:

1. Collection and pre-processing data.
2. Data modeling.
3. Content - based filtering method.

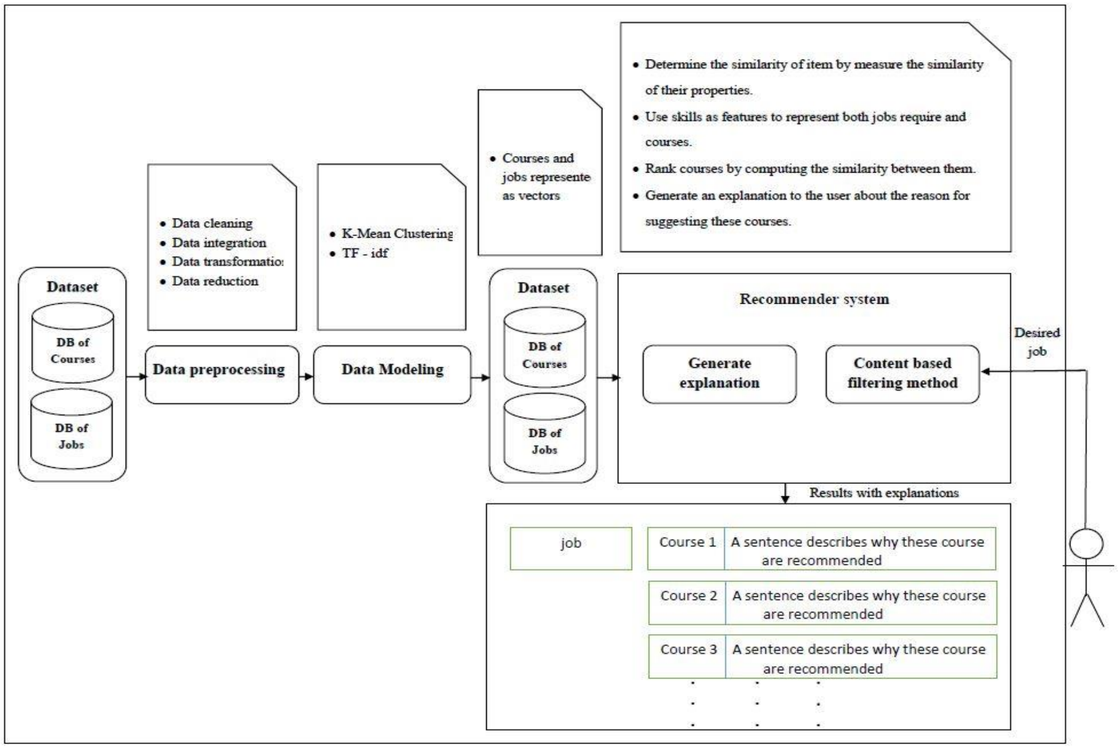

Dataset
DB of Courses
DB of Jobs
Data cleaning
Data integration
Data transformatio
Data reduction
Data preprocessing
K-Mean Clustering
TF - idf
Data Modeling
Courses and jobs represente as vectors
Dataset
DB of Courses
DB of Jobs
Determine the similarity of item by measure the similarity of their properties.
Use skills as features to represent both jobs require and courses.
Rank courses by computing the similarity between them.
Generate an explanation to the user about the reason for suggesting these courses.
Recommender system
Generate explanation
Content based filtering method
Desired job
Results with explanations
job
Course 1
A sentence describes why these course are recommended
Course 2
A sentence describes why these course are recommended
Course 3
A sentence describes why these course are recommended

Figure (1) system architecture

4.1 Collection and Pre-processing of Data

4.1.1 Dataset

In this research, a dataset from courses and jobs was used, which we will describe below.

4.1.2 EDX Database Courses

The EDX dataset contains (397) computer science lessons, which are in English, and include a short description, long description (About), lesson objectives (What you'll learn), Headlines (Syllabus), and also language, level, course length, course hour, and course page link. The Name and short description, long description (About), and lesson objectives (What you'll learn) were used in the experiments of this study. We merged the (Description, About, and What You'll Learn) to become a description for each course. Then, we used the K-Means clustering algorithm to cluster the course dataset into 12-course clusters based on expert opinion. Next, we merged the descriptions of each course cluster and renamed them based on their new contents. To do this properly, we had to convert the file format from Excel into the text because the one cell in Excel does not accommodate more than 50,000 characters. At last, we have a set of words to describe each course cluster.

4.1.3 SEEK Database Jobs

This database contains 14795 job postings registered on the SEEK website, which are in English and include a (Job Listing Date, Location, Salary, Work Type, Classification, URL, and Requirements). The title job and the requirements were used in the experiments of this study. When reviewing the jobs database, we found many duplicate advertisements, so based on that, we used the duplicate advertisements factor in such a way as to indicate the importance of the job, also, we found the advertisements for the post doctor research fellow job is significantly more than the number of the advertisements for other jobs. In addition, the cloud job also contains a large number of advertisements, which indicates the importance of these jobs in today’s market also, the database contains short and unhelpful advertisements. Therefore, it needs to be pre-processed. After pre-processing the database, we chose the requirements field to become a description for each job. Next, we used the K-Means clustering algorithm to cluster the jobs dataset into 18 job clusters based on their descriptions. We set the number of clusters based on expert opinion. Next, we merged all descriptions of each job cluster and renamed them based on their new contents. To do this properly, we had to convert the file format from the Excel file into the text file because the one cell in Excel does not accommodate more than 50,000 characters. Finally, we have a set of words to describe each job cluster.

4.1.4 Pre-processing Data

We decided to use the Python language in our research because it provides flexibility in data processing and application of algorithms through the presence of a group of libraries such as Pandas, Sky-Learn, and other libraries that are used in this research. Depending on the data set used, it becomes important to apply the processing data step between the two datasets to prepare the data to apply the second step in our approach. The second step is to cluster our datasets and then apply the TF-IDF. So we need to preprocess the dataset correctly to get good results. Processing dataset includes (data cleaning, data integration, data transformation, data reduction) data cleaning include changes such as (removing blank and trailing spaces, converting uppercase to lowercase, deleting numbers, removing punctuation., removing stop words, etc.) The important point of these steps is that due to the special nature of this research, special attention must be paid to the sequence of steps so that useful information is not mistakenly deleted.

4.2 Data Modeling

After collecting and preparing the dataset, we use the K-Means clustering algorithm to cluster our dataset into sets of job clusters and course clusters. Then we will have a set of words to describe course clusters and job clusters. We map these words to the vector space model using word embedding algorithms by calculating the tf-idf for our dataset. We can do that in two ways (mathematically or using the sky-learn library). If we calculate the tf-idf mathematically, we should calculate BackOfWord for our dataset to extract the unique words and put them in a dictionary for comparing these words with our dataset. Then we will be able to calculate the term frequency (tf) to know how often a term occurs in a document. Next, we calculate inverse term frequency (idf), which decreases the weight for commonly used words and increases the weight for words that are not used very much in a collection of documents. Finally, we used (tf) and (idf) to calculate (tf- idf) to know how relevant a word is to a document in a collection of documents. Or we can use the skylearn library directly to calculate tf-idf as we do in our approach.

4.3 Content - Based Filtering Method

After getting the tf-idf for our dataset, we will use CBF because the Content-based filtering algorithm deals with items and focuses on their properties. Also, the similarity between them is determined by measuring the similarity in their properties. The main idea of our system is to use the skills as features. So we used the skills provided by the jobs companies as a condition for employment as features to represent jobs. In addition, we used the skills provided by the courses as features as well. Then we used these jobs and courses under the vector space model to represent them as vectors in a high dimensional space. As a result, each vector will correspond to a term. Given job and course vectors, one for each job and one for each course, we used three different measures to rank courses by computing the similarity between the required job and all courses to find the related courses. Also, we generate an explanation to the user about the reason for suggesting these courses.

5. System Evaluation

As we discussed in the previous section, the main base of our approach is a word-based approach that processes a dataset to provide a list of words with their scores. Then, word-based similarity methods are used to find the similarities between the required job vectors and the course vectors to suggest a list of related courses. Several experiments were used to evaluate the quality of the results. In addition, many attempts were made to maximize the quality of outputs by overcoming existing challenges. Below is a description of the experiments that provided the best results, Plus the experiments didn't obtain the desired results. Finally, these results are evaluated based on the results of experts.

5.1 Our Experiments

The EDX dataset was used, which included 12- clusters of courses. In addition, the SEEK dataset was used which contains 18 job clusters. which has already been preprocessed before. We used three different measures. In the first phase, the Cosine Similarity and the Euclidean distance measures were used. Based on that, the two datasets were processed to generate a vector by calculating the TF-IDF score for all words in all documents and creating an array in which each row was a vector for a single document. Finally, the Cosine Similarity and Euclidean distance measurements were calculated between the required job and all courses in our EDX dataset to generate a list of courses related to the selected job. Table 1 shows five jobs and the related courses for each one according to their scores. As can be seen from the column related to the cosine similarity measure, higher-ranked courses are more relevant to the job. For the column that related to the Euclidean distance measure, the lower-ranked courses are more relevant to the job. That indicates good data quality and the appropriateness of the model parameters. These results indicate that the proposed approach can be a solution to the problem of this research. In the second phase, the Jaccard similarity measurements were applied. and the EDX dataset was used, which included 12- clusters of courses. In addition, the SEEK dataset was used which contains 18 job clusters. which has already been preprocessed before. Then, the two datasets were tokenized to generate a set of tokens for all words in all documents. Finally, the Jaccard similarity measure was used to calculate the similarity between the required job and all courses in our EDX dataset to generate a list of courses related to the selected job. Table 1 shows five jobs and the related courses for each one according to their scores. The purpose of this experiment was to extract the similarity using the Euclidean distance measure. Also, this result indicates that the proposed approach can be a viable solution to the problem of this research. In the expert evaluation section, these results will be compared with the experts' results and the quality of the results will be calculated with quantitative criteria.

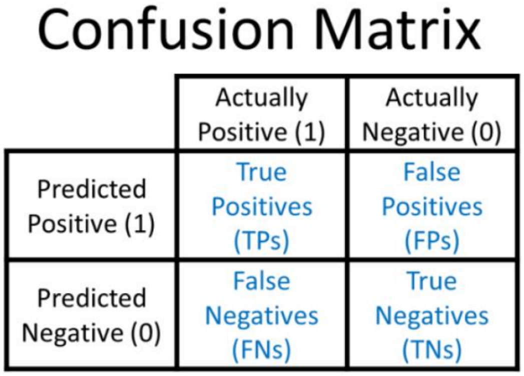
Table (1) shows five jobs and the related courses for each one according to their scores based on different measures

Related courses based on Experts' results	**Cosine similarity measure**	**scores**	**Euclidean distances measure**	**scores**	**Jaccard coefficient measure**	**score**
Post Doctor Research Fellow job						
Machine Learning	Software	0.39119161	Software	1.10345674	Software	0.168082192
Big data	Big data	0.35954089	Big data	1.13177657	Data Science	0.15130674
Software	Programming	0.30893092	Programming	1.17564372	Programming	0.146932558
Programming	Digital Media	0.29340903	Digital Media	1.18877329	Cloud Computing	0.136640798
Data Science	Cybersecurity	0.26162625	Cybersecurity	1.215215	Cybersecurity	0.126189144
Cloud job						
Cloud Computing	Software	0.35351996	Software	1.13708402	Software	0.222064202
Programming	Big data	0.31208123	Big data	1.17296101	Data Science	0.198935735
Software	Cybersecurity	0.30478929	Cybersecurity	1.17916132	Programming	0.196102819
Cybersecurity	Cloud Computing	0.29167404	Cloud Computing	1.19023187	Cloud Computing	0.192622092
Big data	Digital Media	0.2804094	Digital Media	1.19965878	Cybersecurity	0.177045696

Senior .net Developer job						
Programming	Software	0.3174602	Software	1.16836621	Cloud Computing	0.29037704
Software	Big data	0.2751485	Big data	1.20403613	Cybersecurity	0.277446301
Web Programming	Web Programming	0.23701038	Web Programming	1.23530532	Big data	0.267029973
Data Structure and Algorithm Design	Cybersecurity	0.22593429	Cybersecurity	1.24423929	Software	0.262295082
Cybersecurity	Programming	0.22485606	Programming	1.24510557	Programming	0.254891592
Full Stack & Software & C# developer job						
Programming	Web Programming	0.38196607	Web Programming	1.11178589	Software	0.299693016
Software	Software	0.31543928	Software	1.17009463	Cloud Computing	0.283857442
Web Programming	Programming	0.27779785	Programming	1.20183372	Programming	0.27464503
Data Structure and Algorithm Design	Big data	0.26265999	Big data	1.21436404	Data Science	0.272154391
Big data	Cloud Computing	0.22059729	Cloud Computing	1.24852129	Cybersecurity	0.263020833
Software .net developer job						
Programming	Programming	0.05633285	Programming	1.37380286	Digital Media	0.019047619
Software	Software	0.05144057	Software	1.37735938	Digital Marketing	0.018348624
Web Programming	Big data	0.05002062	Big data	1.37838992	Video Game	0.014975042
Data Structure and Algorithm Design	Digital Media	0.04903943	Digital Media	1.37910157	Web Programming	0.012235818
Cloud Computing	Cybersecurity	0.04325067	Cybersecurity	1.38329269	Cybersecurity	0.012115564

5.2 Evaluation of Program Outputs Based on Experts' Results

To evaluate our results, we created a Google form that contains 18 jobs and 12 courses. Then, we sent it to the experts to select the relevant courses based on the required job. We sent the Google form to many professors in universities specializing in computer engineering or computer science in general, in addition to our friends who are master's and doctoral students, but unfortunately, we received very few evaluations from the professors, which forced us to resend the evaluation form to other students. We evaluated our results based on the received results from experts. We applied a confusing matrix. Fig (2) shows the details of the confusion matrix. Then, we calculated the precision, recall and F- score of our results according to 3 thresholds (2,3, and 5).



Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

Figure (2) the confusion matrix

Finally, we compared our results with the experts' results. The components of the standards were defined as follow:

- TP: Associated courses and included in the output.
- TN: Unrelated courses and did not appear in the list of outputs.
- FP: Unrelated courses but included in the list of outputs.
- FN: Relevant courses but not included.

The formula for precision, recall and F-score is as follows:

$$\boldsymbol{Precision} = \frac{\boldsymbol{True\ positive}}{\boldsymbol{True\ positive + False\ positive}} \qquad \textbf{(1)}$$

$$\boldsymbol{Recall} = \frac{\boldsymbol{True\ positive}}{\boldsymbol{True\ positive + False\ Negatives}} \qquad \textbf{(2)}$$

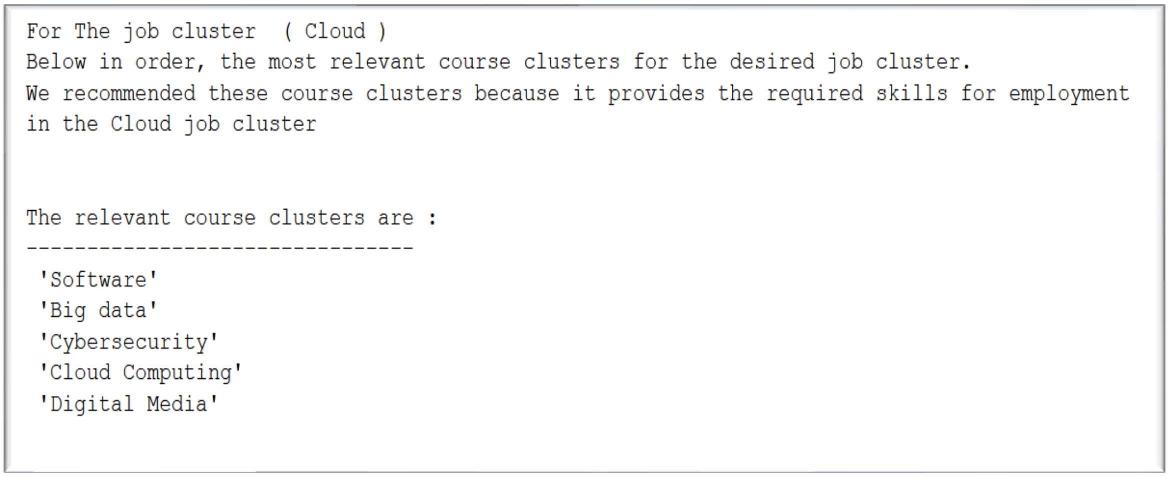
$$\boldsymbol{F1 = 2 * \frac{Precision * Recall}{Precision + Recall}} \qquad \textbf{(3)}$$

to select the appropriate threshold, we used experts' results as a basis to choose the threshold. We made a comparison between the pre-defined thresholds according to the F-score. The best result was when choosing the threshold equal to 5.

Table (2) shows the characteristics of the described experiment.

Dataset of courses	Dataset of jobs	Method	F- score
EDX subset	SEEK subset	TF-IDF and Content based filtering using cosine similarity measure	0.68
EDX subset	SEEK subset	TF-IDF and Content based filtering using Euclidian distance measure	0.68
EDX subset	SEEK subset	Content based filtering using Jaccard coefficient	0.52

Table (2) shows the characteristics of the described experiment. When comparing our results with the results of experts, it turns out that the approach used in the first and second experiments is better than the approach used in the third method. When we compared our results for the jobs (cloud, Senior .net Developer, and Full Stack & Software & C# developer jobs) with the experts' results, the percentage was 80%. It is a good percentage. Also, for the job (Post Doctor Research Fellow), when comparing our results with the experts' results, the percentage was 60%. That does not mean that the quality of the data is low, but the reason may be the inconsistency in expert assessments. Finally, when evaluating the job of (Software .net developer job), we found that the percentage is 40% compared to the results of the experts. We investigated that matter it became clear that there were very few advertisements for this job compared to other jobs. In addition, the advertisements for this job contained a repetition of 90% were removed during the stage of processing data. It shows, that if the number of advertisements for a job is small, that job will be taken as a starting point and the probability of providing a suitable list of relevant courses will decrease. However, despite the little data on this job, the system recognizes some similarities and the output provided is not completely irrelevant. The results for each job were evaluated by calculating the precision, recall and the F- score and then the average. The mean F-score was 68% for the TF-IDF approach using cosine similarity and Euclidean distance measures. As for the approach Jaccard coefficient the percentage was 52%. The reason for this is due the Jaccard coefficient approach doesn't consider term frequency (how many times a term occurs in documents). Also, this approach doesn't consider rare terms. The rare term in a collection is more informative than the frequent term. Finally, in Figure (3) we are showing a sample of our system's output which shows the desired job and the recommended courses with an explanation of why these courses are recommended. The courses were recommended based on the similarity degree between the required skills for employment and the skills offered by courses, the higher-ranked courses are more relevant to the job.



```
For The job cluster  ( Cloud )
Below in order, the most relevant course clusters for the desired job cluster.
We recommended these course clusters because it provides the required skills for employment
in the Cloud job cluster


The relevant course clusters are :
--------------------------------
 'Software'
 'Big data'
 'Cybersecurity'
 'Cloud Computing'
 'Digital Media'
```

Figure (3) showing the desired job and the recommended courses with an explanation of why these courses are recommended

6. Conclusion

In this project, we presented a course recommendation system that uses the K-Mean Clustering algorithm, TF-IDF approach, and content-based filtering algorithm to recommend the related courses based on the desired job, with an explanation of why these courses are recommended. We tried to discover the relationship between job opportunities and training courses by discovering the available texts on the web and then using different methods to check the presence of this relationship. Based on the results of this study, the quality of the data is the most important factor and has the greatest impact on the quality of the outputs. If the size of the data is sufficient, then the proposed methods will give the desired outputs and the relationship between courses and jobs can be extracted from the texts using the machine and implement the processes of text discovery. Based on our data, we noticed when the number of query terms is big the algorithm is better at predicting. Regarding our system, the expected question is about the benefit of the system and whether students will benefit from using it and get the required job. We need to follow up on the students' jobs who used the system based on its outputs in developing their career plans, so we need a long time. In theory, students who use personalized course recommendations to achieve the desired job will get work in related fields faster than students who do not benefit from the system recommendations. From a workable perspective, there are many ways in which our program can be improved. One of these ways, use more data for courses than the currently used. In addition, periodically update the data used to ensure that it is fit for purpose. Regarding this, the model will remain relevant over time and this will be a low-cost task because our solution is unsupervised. In conclusion, the proposed system is novel and offers many advantages compared with other recommender systems. our system converts a simple course recommendation into a tool for discovering skills. Based on that, the system will recommend the relevant courses. Since many recommendation systems work as black boxes, it is difficult for students to understand why these courses are recommended. So, we designed our system to recommend the relevant courses with explain why these courses are recommended. This will add a factor of transparency to our system and confirm the reliability of the system to the students. therefore, our system will enable students to make the right decision about their future job plans which will lead them to reach their dream job.

References

[1] B. Ma, “Course Recommendation for University Environments,” Proc. 13th Int. Conf. Educ. Data Mining, EDM 2020, vol. 1, no. Edm, pp. 460–466, 2020.

[2] B. Mondal, O. Patra, S. Mishra, and P. Patra, “A course recommendation system based on grades,” 2020 Int. Conf. Comput. Sci. Eng. Appl. ICCSEA 2020, 2020, doi: 10.1109/ICCSEA49143.2020.9132845.

[3] Adomavicius, Gediminas, and Alexander Tuzhilin. "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions." IEEE transactions on knowledge and data engineering 17.6 (2005): 734-749.

[4] Geetha, G., Safa, M., Fancy C., Saranya D. "A hybrid approach using collaborative filtering and content based filtering for recommender system." Journal of Physics: Conference Series. Vol. 1000. IOP Publishing, 2018.

[5] P. V. Kulkarni, S. Rai, and R. Kale, “Recommender System in eLearning: A Survey,” no. January, pp. 119–126, 2020, doi: 10.1007/978-981-15-0790-8_13.

[6] Nallamala, Sri Hari, et al. "A Brief Analysis of Collaborative and Content Based Filtering Algorithms used in Recommender Systems." *IOP Conference Series: Materials Science and Engineering*. Vol. 981. No. 2. IOP Publishing, 2020.

[7] P. Melville and V. Sindhwani, “Encyclopaedia of Machine Learning: Recommender Systems,” *Encycl. Mach. Learn.*, pp. 829–838, 2010, [Online]. Available: https://link.springer.com/content/pdf/10.1007/978-1-4899-7687-1_964.pdf%0Ahttps://link.springer.com/referenceworkentry/10.1007%2F978-0-387-30164 8_705%0Ahttps://link.springer.com/book/10.1007/978-0-387-30164 8%0A%0Ahttp://vikas.sindhwani.org/recommender.p.

[8] Gönül, M. Sinan, Dilek Önkal, and Michael Lawrence. "The effects of structural characteristics of explanations on use of a DSS." *Decision support systems* 42.3 (2006): 1481-1493.

[9] J. Daher, A. Brun, and A. Boyer, “A Review on Explanations in Recommender Systems To cite this version : HAL Id : hal-01836639 A Review on Explanations in Recommender Systems,” 2018.

[10] B. Behkamal, B. B. Haghighi, and A. T. Haghighi, “A heuristic method for curriculum planning based on students ’ interest,” pp. 19–21, 2019.

[11] Z. A. Pardos and W. Jiang, “Designing for serendipity in a university course recommendation system,” *ACM Int. Conf. Proceeding Ser.*, pp. 350–359, 2020, doi: 10.1145/3375462.3375524.

[12] A. Esteban, A. Zafra, and C. Romero, “Helping university students to choose elective courses by using a hybrid multi-criteria recommendation system with genetic optimization,” *Knowledge-Based Syst.*, vol. 194, p. 105385, 2020, doi: 10.1016/j.knosys.2019.105385.

[13] G. Zhu, Y. Wang, K. Jona, N. A. Kopalle, X. Liu, and K. Börner, “Community-based data integration of course and job data in support of personalized career-education recommendations,” *arXiv*. 2020, doi: 10.1002/pra2.324.