TELECOM PARIS

DATA MINING PROJECT
SD 201

Coursera Course Recommendation System

Group :
Ghada Ben Ammar
Hajer Ben Ammar
Mayssa Haddar
Oussama Zaibi

Contents

1 Introduction

1.1 General context

Over the last decade, online learning has grown significantly as the internet and education have combined to provide people with the possibility to master new skills. Since the COVID-19 epidemic, people's lives have been more centered on online learning. The pandemic has caused schools, universities, and businesses to operate remotely, which has increased the use of online learning. Even before the pandemic, Research and Markets predicted that the online education market would be worth $350 billion by 2025, therefore the figures may be revised after studying the growth consequences of COVID-19 on the online learning market[6].

There are various online learning platforms in the market that serve millions of individuals, including Udemy, Coursera, Lynda, Skillshare, and Udacity. Different user verticals are also shaping the platforms. While Skillshare caters to creatives by offering courses in animation, photography, and lifestyle, Coursera caters to academics by providing access to university courses. And that's where comes the need for the recommendation systems and their importance.

In the e-learning area, a content recommender system assists learners by recommending appropriate learning resources based on their interests and learning goals.

1.2 Objectives

Our project aims to help any new learner get the right course to learn given his preferences. Our goal is to answer the following questions :

- Can we get an accurate course recommendation system based on a few features of courses?
- Would it be better if the user introduced as input a list of keywords instead?
- Can we create a more accurate rating system for courses ?

2 Data preparation

2.1 Dataset: Overview

2.1.1 About the dataset

The project's dataset "Coursera courses and reviews" contains mainly a maximum number of available courses on Coursera along with their reviews and ratings.

- The data is collected from 2015 till 2020.
- The total number of attributes is 22.
- The total number of reviews is 420441.
- The total number of courses is 469.
- The total number of institutions is 116.

2.1.2 Data columns and description

The final data is presented in table 1. In fact :

- The last column indicates whether the feature was already in the initial data or added later on in the feature engineering part.
- There are two types of general ratings. The first one is the overall rating of the course which is given by Coursera. The second one is the `rating_mean` which is calculated using the ratings of the reviewers themselves for each course.

Column	Description	Original/Added
course id	Course's id	Original
name	Course's name	Original
institution	Name of the institution offering the course	Original
course url	Course's url	Original
Course Description	A paragraph describing the course's content	Original
Skills	It lists the skills you will acquire when you finish the course	Original
Difficulty Level	Difficulty level of the course (Beginner, Intermediate, Conversant, Advanced)	Original
Course Rating	The overall course's rating given by coursera	Original
reviews	Reviews given by learners	Original
reviews clean	Clean reviews containing filtered and specific words only	Added
number words	Number of words in the cleaned reviews	Added
reviewers	Name of the reviewer	Original
rating	Rating of the course given by the reviewer	Original
rating mean	Average rating of the course calculated from reviewer's ratings	Added
rating count	Total number of reviews for each course	Added
year, month, day	Date of the review	Modified
neg, pos, neu, compound	Features extracted from reviews' sentiment analysis	Added
weighted score	Weighted score made by the IMDB method	Added
Combined feature	Combine multiple string features from the dataset	Added

Table 1: Data columns and their description

2.2 Data pre-processing and feature engineering

2.2.1 Data preprocessing

In this part, we went through 4 steps in order to get our data ready and clean.

- Our dataset is actually the result of merging two different datasets [2] and [1] : the first one contains information and details about the courses of Coursera and the second one has more details about Coursera's course reviews. So we did the merge on course ids and we dropped the duplicated columns.
- For our second step, we eliminated the rows with no reviews so that the course ratings would have better meaning and importance in our recommendation system.
- We also removed the duplicated reviews.
- And lastly, the column that was initially named `date_reviews` was divided into three columns 'year, 'month' and 'day'.

2.2.2 Feature engineering

As our objective is to generate a recommendation system, creating and manipulating variables is a good step to achieve better results.

- We created two new features : `rating_count` provides information on the number of reviews received by a certain course, which can provide insight into the overall number of people who have completed the course and `rating_mean` computes the mean review of the course by the reviewers in our dataset (Which differs from course rating that is the overall rating of the course in coursera) These two characteristics, together with the year taken from `'date_reviews`, will be utilized to construct an IMDB style rating for each course.
- We created another feature which is `reviews_clean` based on the given review : Ordinarily, when using text or NLP, we do not have nice, easy-to-use keyword lists. In that case, there are a number of standard, simple steps to take in order to essentially extract a clean text. They are: Lowercasing words, lemmatizing words, removing punctuation , stop words, empty tokens, words with only one letter and words with numbers in them. These can all be accomplished simply by using the nltk library.

- We also created the compound, neutral, positive and negative features for each review : We extracted the overall sentiment in each review in our dataset, so we could tell not only whether a review is positive or negative, or something in between (neutral) but also the level of negativity/positivity in said comment. This will be obtained by the Compound Feature that ranges from -1 to 1, -1 being the maximum negative value and 1 being the maximum positive value. This feature will later be used to filter recommended options (we recommend only courses with positive sentiment so mean compound over a certain threshold)

2.3 Data comprehension and analysis

This step consisted of making different visualizations in order to better understand our data and to get an overview of the content.

- At first, we were interested in the distribution of the reviews based on the dates (month and year). We noted that the "pic" of reviews for each year occurs in April, May, October, and December. These times correspond to the end of each semester, which is test preparation time. Students frequently look for courses that are similar to what they have so that they can revise by using them. We also noticed that the year 2020, which coincides with the Covid pandemic, has the most reviews, implying that the learning rate has increased that year due to the availability of learners.

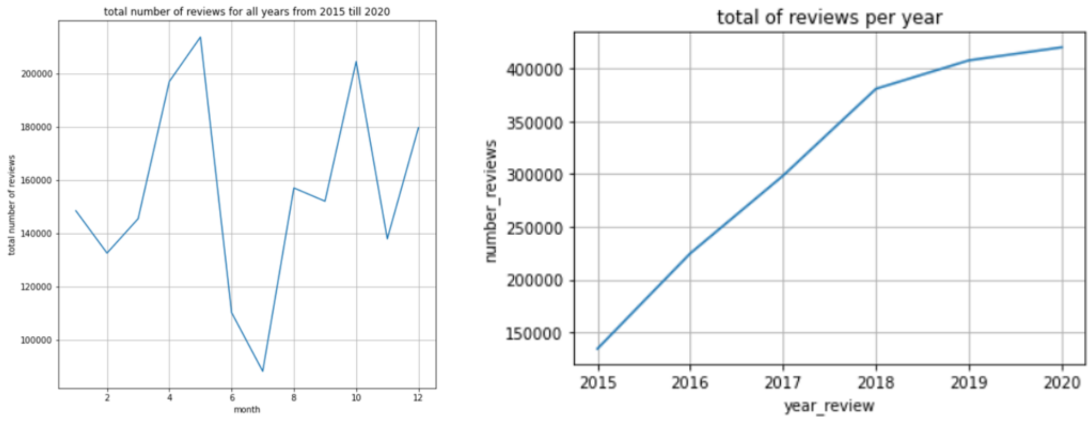


Figure 1: Total number of reviews (a) in each month for all years together and (b) per year

- Then, we looked into the distribution of courses based on their difficulty level and also the combination of the difficulty level and course rating. And we finished by plotting them based on the reviewers' ratings.

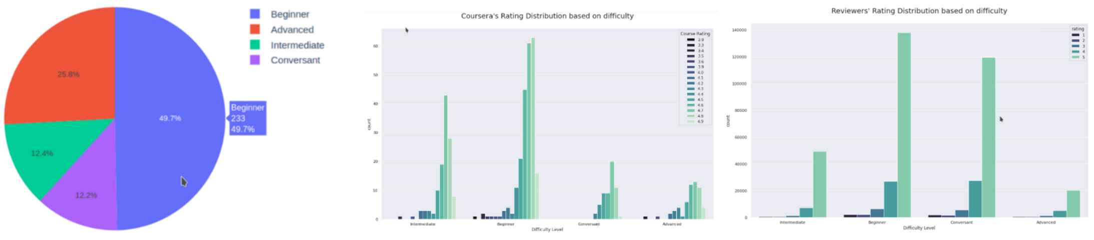


Figure 2: The distribution of courses based on (a) difficulty level (b) difficulty level and course rating (c) difficulty level and reviewer's ratings

We can say that most of the courses that coursera offers are "Beginner" and "Advanced" levels which makes the platform beginner friendly and accessible for everyone. The most top rated courses by

Coursera are mainly beginner and intermediate courses based on the plot (b). However, in the plot (c) we can see that the conversant courses are much better rated by the reviewers than the intermediate ones. This result makes us wonder : is there a correlation between ratings of Coursera and the average rating of reviewers? We will answer this question in the next point.

To answer this question, we chose one the best and worst rated courses and we plotted the distribution of reviewers' ratings and the overall rating of Coursera and we compared them. We can see that there is a strong correlation between these two features for the best rated course. However, the reviewers' ratings are scattered around the red line which represents the overall rating of the worst rated course.

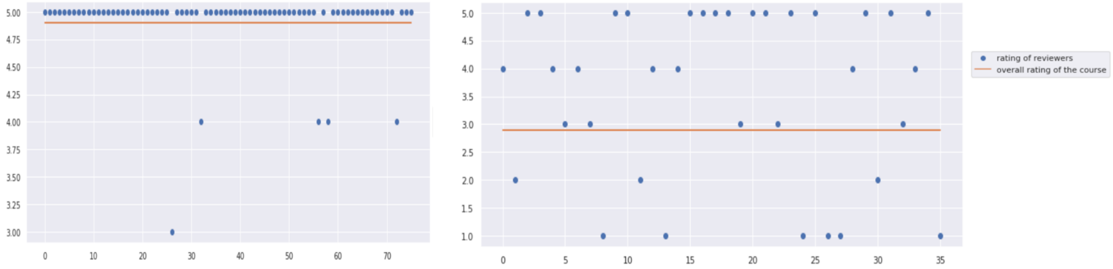


Figure 3: Correlation between reviewers' ratings and Coursera's global ratings for (a) the best and (b) the worst rated courses

- We then looked at the number of courses offered by each institution. We found that the vast majority of schools offer between 1 and 5 courses.

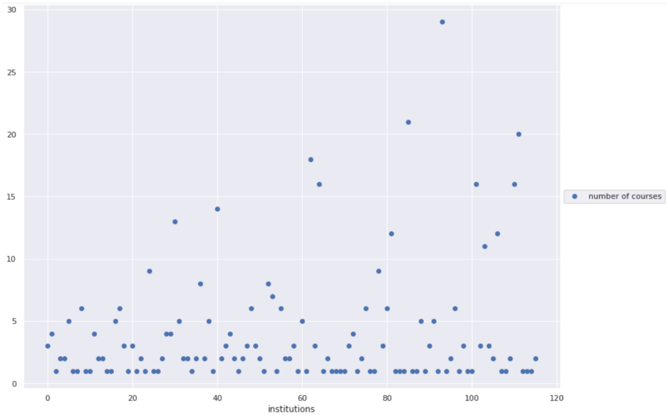


Figure 4: The number of courses for each institution

Hypothesis: The university with the biggest number of courses is among the top 5 best rated courses. –> We found that the "University of Houston" is the one that offers the largest number of courses. However, it is not one of the top rated universities. Its average rating is 4.6 which is not bad but lower than the best rating which is 4.9.

2.4 IMDB Score generation

This section will try to make a new "fair" way of rating courses. In fact, using the average ratings of a given course is not the best decision to make since a course with only 3 votes and a 4.8 average rating, for example, is not necessarily better than another with 4.0 as an average rating with 100 votes.

To do so we'll be using the IMBD's weighted rating which is given by the following equation:

$$WeightedRating(WR) = (\frac{v}{v+m}.R) + (\frac{m}{v+m}.C)$$

where:

- v is the number of votes for the course.
- m is the minimum votes required to be taken into consideration.
- R is the course's average rating.
- C is the mean vote across the whole dataset.

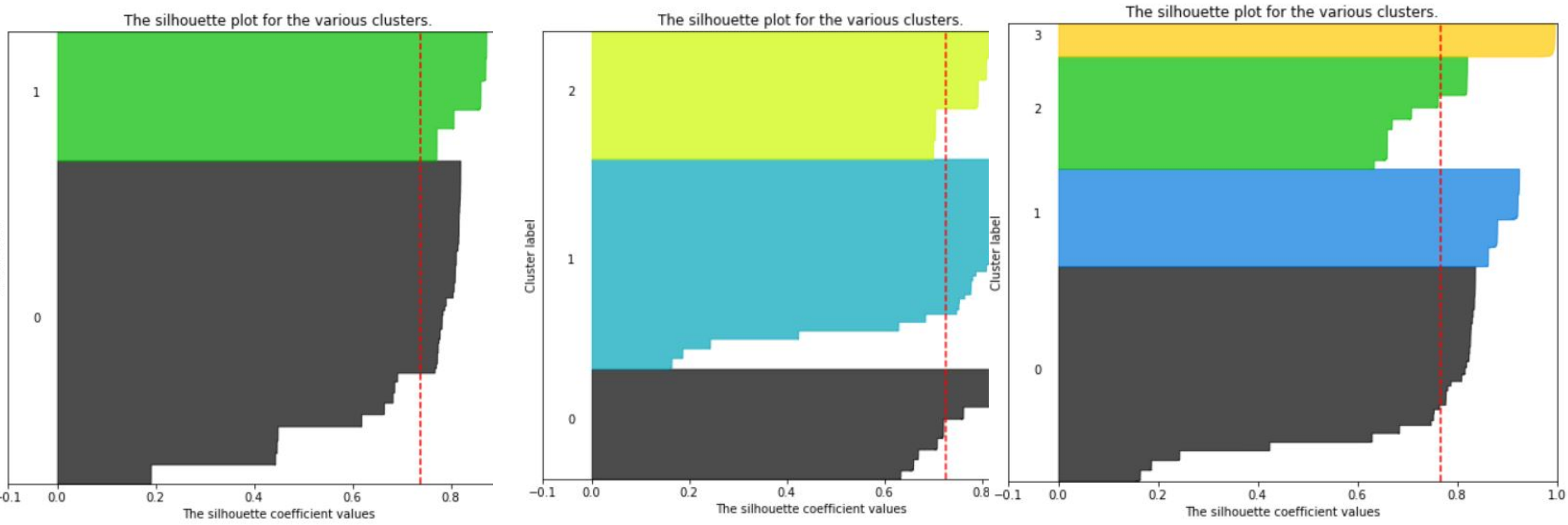
As you may have noticed, only a fraction of our data will get an IMDB score, so we will try an experiment and predict for the rest of our dataset the missing IMDB scores using two different approaches: unsupervised learning with classification and regression.

2.4.1 First approach

The first approach will simply be a regression model applied to the IMDB column to predict null values. We will be using as a model the catboost regressor. Results of the inference will be discussed in the next section.
About catboost: CatBoost is a gradient-boosting algorithm on decision trees developed by Yandex researchers and engineers[3].

2.4.2 Second approach

The second approach is the most interesting, maybe because it combines both unsupervised and supervised learning to predict the missing IMDB scores.
We will first determine the number of classes in question. We will use for this unsupervised phase the algorithm K-means and detect the best number of clusters via silhouette analysis.
For a good number of clusters estimation, all clusters need to have a silhouette value above the average silhouette score as well as to be equi-distributed between classes[5]. In other words, the clusters need to be more or less of similar thickness and hence proportional class sizes. Results are shown in the following figure:



((a)) $clustersnumber = 2$ ((b)) $clustersnumber = 3$ ((c)) $clustersnumber = 4$

Figure 5: Silhouette plots for various class numbers

We will then use a classifier to detect the class of each course's feedback that doesn't have an IMDB score yet. Then assign to this row the mean of the cluster.

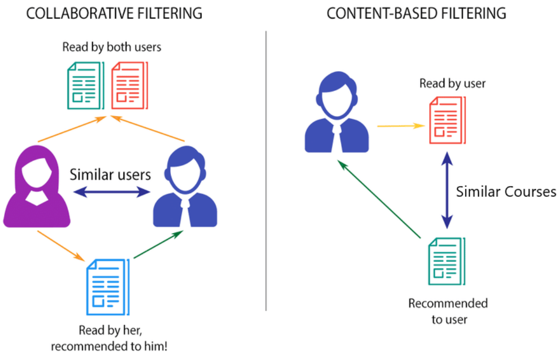
However, a course without an IMDB score will have different comments which means multiple rows hence multiple IMDB scores are predicted. For this final step, we will simply make the mean of scores per course.

3 Recommendation system implementation

3.1 Content-based recommendation systems

Many big platforms use recommendation systems. These recommendations could range from products to services and even to other users' profiles. The different use case scenarios and the goal of each platform should be taken into consideration while choosing the type of recommendation system to utilize:

- Collaborative Filtering Systems: These systems take into account other users' data concerning different products to make a recommendation for a certain user.
- Content Based Systems: This type of recommendation systems makes recommendations based on the product itself and the preferences of the user. [4]



For this project, we will be creating a Content Based recommendation system after taking into consideration the type of data we have:

Our data set has Item level information (Course name, description, difficulty level, etc...) and user level information (User feedback on courses: rating, reviews, etc...) which makes it eligible for the content-based recommendation type.

3.2 Model

We chose to work with cosine similarity: Cosine similarity is a measure of how similar two sequences of numbers are. It calculates the normalized dot product of two vectors X and Y.

In the case of two data sets, it will output the cosine similarity between samples of X and Y. Cosine similarity is calculated as follows for two vectors X and Y :

$$K(X,Y) = \frac{< X, Y >}{||X|| * ||Y||}$$

Each cosine similarity ranges from -1 to 1:

- -1 indicates that the two samples are complete opposites.
- 0 indicates a lack of correlation between the two samples.
- 1 indicates the both samples are identical.

In the context of our project, we will be using cosine similarity to calculate the cosine distance between the vector of the course and the vector of the user to determine the likelihood of said user enjoying that course.
We are going to create 2 different models, based both on different notions of similarity. The first model will use cosine similarity with giving , and the second one will use Jaccard similarity. Within the 2 models, the first will use simple word counts, and the second will use tf-idf to create course vectors.

3.2.1 First Method : Word counts with cosine similarity

3.2.1.1 Implementation

In order to do that, we started off by creating vectors for both the concerned user and the different courses.
We chose to work with a combination of different features for courses that we judged as most relevant: Course name, difficulty level, course description and the skills offered by the course.

We didn't use the institution feature to compute our course vector because it could add faulty bias to the similarity score: Not all courses offered by the same university are of the same topic, difficulty level and end goals. Since the user will be looking for a course most similar to another he liked, using the institution feature could lead to recommendations of courses having different general topics, but sharing the same institution as the preferred course.

After combining the useful features, we used Count Vectorize to create a matrix containing the vector of each course.
Count Vectorizer is a method to convert text, which is the type of our combined feature, into a numerical vector so that it could be comprehended by our model. It creates a vector based on the number of occurrences of each word in the text.

Once we obtained our courses matrix, we calculate the pairwise similarity of all elements in the matrix by taking Y=null. We will then use the resulting matrix, to retrieve all Similarities with the preferred course of the user. We then sort the similar courses and recommend the top 10 similar courses to the user.

3.2.1.2 Post-filtering

- Goal :
While content-based recommendations are a great way to output similar courses in terms of difficulty and learned skills, it doesn't take into account the rating of the course and the opinions of other users on that course.

The solution we found is to do a post-filtering on similar courses, to filter out courses that have mostly bad reviews and bad general sentiment. In order to make the filtering the most accurate possible, we chose to work on two different features:

- IMDB Style Score:
  We created a feature based on the IMDB platform scoring system. This score takes into account not only the overall rating of the course but also the number of reviews per course.
- Sentiment Analysis Compound score:
  Sentiment analysis compound score associates a score ranging from -1 to 1 to a text, in our case a user's review, -1 being the most negative score and 1 being the most positive score. Taking into

account the sentiments of the reviews is important because they could provide a more detailed opinion on the course than a rating does.

- Sentiment Analysis based filtering:

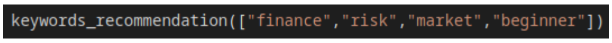
As our data contains reviews from different users for different courses, we made use of that fact to extract the sentiments of the reviews and utilize them to filter out the negatively rated courses.

For that, after having cleaned our review data, we proceeded to work with nltk sentiment intensity analyzer [8]. We then used the mean compound feature to filter out negatively viewed courses, with a mean compound score lower than 0.

3.2.2 Second Method: tf-idf with cosine similarity

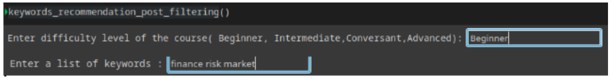
3.2.2.1 Implementation

In this method, we chose to use the tf-idfVectorizer[7] because while the CountVectorizer performs the task of tokenizing and counting, Tf-idfVectorizer performs all three operations (tokenizing, counting and normalizing), thereby streamlining the process of NLP.
Just like we did in the fist method, we start by selecting the columns we will need which are : Course name,course description,skills, difficulty level.
Since we are creating **keyword-based recommenders** , we need the keywords column: WE wanted to see if our recommendation system provides different results when our "keywords" column contains Course name,course description, skills and the difficulty level (figure 6) and when the "keywords" column doesn't contain the difficulty level. In the second case, we filter the obtained result with the chosen difficulty level afterwards (figure 7), all while operating within the same recommendation system.
The keywords column is created by combining the chosen columns(for each case) and being cleaned using the nltk library. (Check the notebook for more details about the steps)



```
keywords_recommendation(["finance","risk","market","beginner"])
```

Figure 6: input for the first recommender system



```
keywords_recommendation_post_filtering()
Enter difficulty level of the course( Beginner, Intermediate,Conversant,Advanced): Beginner
Enter a list of keywords : finance risk market
```

Figure 7: input for the second recommender system

3.2.2.2 Post-filtering

This approach is effective if the researcher knows exactly what they are looking for. The problem is words often have multiple meanings, so keyword searches often return irrelevant results (false positives), failing to disambiguate unstructured text. An other point is that even though keywords-based recommendations are a great way to output similar courses in terms of difficulty and learned skills, it doesn't take into account the rating of the course and the opinions of other users on that course.

SO just like we did in the first method, we will do a post-filtering on similar courses, to filter out courses that have mostly bad reviews and bad general sentiment. In order to make the filtering the most accurate possible, we chose to work on the mean compound feature.

4 Results and Analysis

4.1 IMDB Score generation results

The IMDB Scoring method has given quite satisfying results when clustered with Silhouette Analysis. We can explore some results in the following figures where the 3 classes were easily discernible.

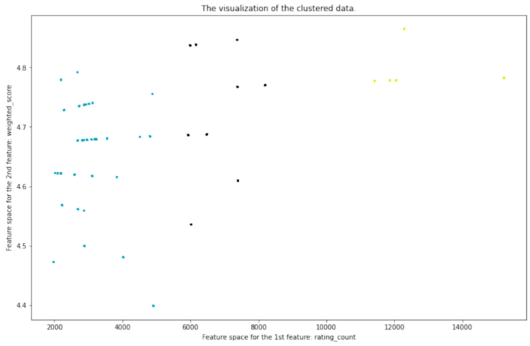


Figure 8: 2D plot of Random columns and the IMDB score

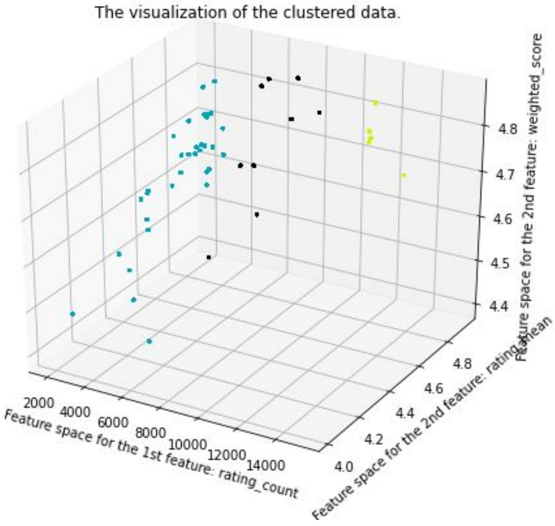


Figure 9: 3D plot of a random column and the IMDB score

However, the classification method failed because the model was overfitting due to the low number of courses used in the learning phase (23 courses only).

As for predicting the IMDB score straightforwardly using regression, we used RMSE as an evaluation metric, which gave us pretty good but suspicious results (RMSE = 1.52e-07) since the target column (IMDB scores) has a variance close to zero. That is why we will not rely on the IMDB scoring system as a solid feature for post-filtering in a future section.

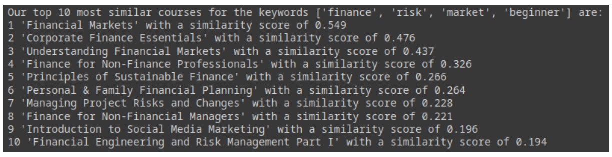
$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

Figure 10: RMSE Formula

4.2 Recommendation system results' comparison

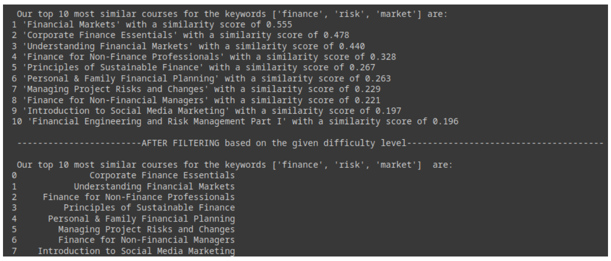
4.2.1 Comparing the two methods used in tf-idf

The first method and the second one (before filtering based on difficulty level) give the same result (the same courses in the same order) with a slight difference in similarity score which is normal because in the second method the `difficulty_level` is not counted. After filtering, we get a better selection of courses based on difficulty level which is optimal compared to the first method.



```
Our top 10 most similar courses for the keywords ['finance', 'risk', 'market', 'beginner'] are:
1 'Financial Markets' with a similarity score of 0.549
2 'Corporate Finance Essentials' with a similarity score of 0.476
3 'Understanding Financial Markets' with a similarity score of 0.437
4 'Finance for Non-Finance Professionals' with a similarity score of 0.326
5 'Principles of Sustainable Finance' with a similarity score of 0.266
6 'Personal & Family Financial Planning' with a similarity score of 0.264
7 'Managing Project Risks and Changes' with a similarity score of 0.228
8 'Finance for Non-Financial Managers' with a similarity score of 0.221
9 'Introduction to Social Media Marketing' with a similarity score of 0.196
10 'Financial Engineering and Risk Management Part I' with a similarity score of 0.194
```

Figure 11: recommended courses using difficulty level in the keywords

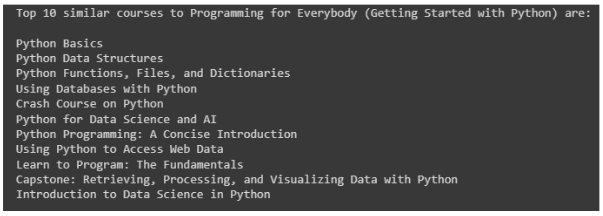


```
 Our top 10 most similar courses for the keywords ['finance', 'risk', 'market'] are:
1 'Financial Markets' with a similarity score of 0.555
2 'Corporate Finance Essentials' with a similarity score of 0.478
3 'Understanding Financial Markets' with a similarity score of 0.440
4 'Finance for Non-Finance Professionals' with a similarity score of 0.328
5 'Principles of Sustainable Finance' with a similarity score of 0.267
6 'Personal & Family Financial Planning' with a similarity score of 0.263
7 'Managing Project Risks and Changes' with a similarity score of 0.229
8 'Finance for Non-Financial Managers' with a similarity score of 0.221
9 'Introduction to Social Media Marketing' with a similarity score of 0.197
10 'Financial Engineering and Risk Management Part I' with a similarity score of 0.196

 -----------------------AFTER FILTERING based on the given difficulty level-------------------------------------

 Our top 10 most similar courses for the keywords ['finance', 'risk', 'market']  are:
0              Corporate Finance Essentials
1           Understanding Financial Markets
2     Finance for Non-Finance Professionals
3         Principles of Sustainable Finance
4      Personal & Family Financial Planning
5        Managing Project Risks and Changes
6        Finance for Non-Financial Managers
7    Introduction to Social Media Marketing
```

Figure 12: recommended courses using difficulty level for post-filtering

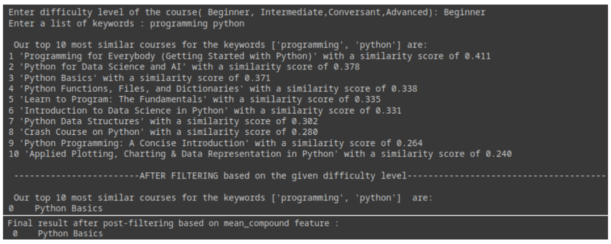
4.2.2 Comparing the results of the words counts and the tf-idf

In this section, we will compare the results of the words counts method and the tf-idf method based on two examples. In the case of words count method, the input is a course name chosen by a user. The model recommends mostly similar courses from the same topic and set of skills. This is especially true for topics that are highly recurrent in our data, like learning python. This could be seen in the first example in figures 13 and 14. The two methods give similar results but the order is better in the tf-idf method. It also gives more general courses and sticks to the input since the user enters keywords and not a course name.



```
Top 10 similar courses to Programming for Everybody (Getting Started with Python) are:

Python Basics
Python Data Structures
Python Functions, Files, and Dictionaries
Using Databases with Python
Crash Course on Python
Python for Data Science and AI
Python Programming: A Concise Introduction
Using Python to Access Web Data
Learn to Program: The Fundamentals
Capstone: Retrieving, Processing, and Visualizing Data with Python
Introduction to Data Science in Python
```

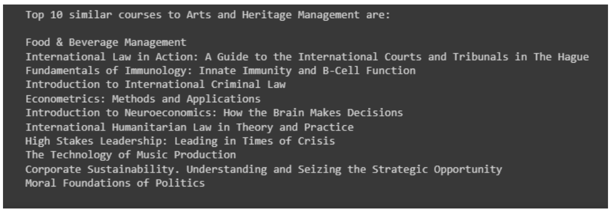
Figure 13: recommended courses using the first recommending system



```
Enter difficulty level of the course( Beginner, Intermediate,Conversant,Advanced): Beginner
Enter a list of keywords : programming python

 Our top 10 most similar courses for the keywords ['programming', 'python'] are:
1 'Programming for Everybody (Getting Started with Python)' with a similarity score of 0.411
2 'Python for Data Science and AI' with a similarity score of 0.378
3 'Python Basics' with a similarity score of 0.371
4 'Python Functions, Files, and Dictionaries' with a similarity score of 0.338
5 'Learn to Program: The Fundamentals' with a similarity score of 0.335
6 'Introduction to Data Science in Python' with a similarity score of 0.331
7 'Python Data Structures' with a similarity score of 0.302
8 'Crash Course on Python' with a similarity score of 0.280
9 'Python Programming: A Concise Introduction' with a similarity score of 0.264
10 'Applied Plotting, Charting & Data Representation in Python' with a similarity score of 0.240

 -----------------------AFTER FILTERING based on the given difficulty level-------------------------------------

 Our top 10 most similar courses for the keywords ['programming', 'python']  are:
0    Python Basics
Final result after post-filtering based on mean_compound feature :
 0    Python Basics
```

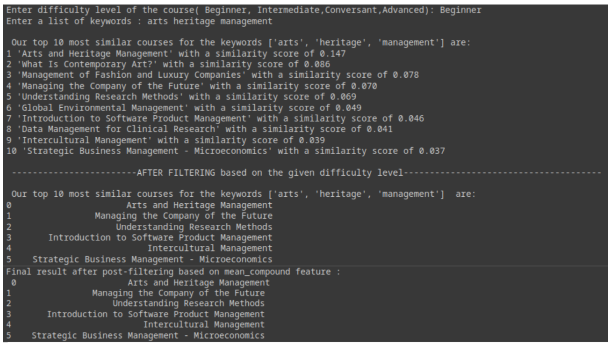
Figure 14: recommended courses using the second recommending system

However, in the second example, we can see in figure 15 that for topics that are not that common in our data set, some recommendations in the words count method are a bit different than the course subject, the similarity in this case could be due to the difficulty level or some shared secondary skills. However, in the tf-idf method, we obtain a much better result. All the recommended courses are related to the keywords and we have the possibility to filter based on difficulty level. (fig16)



```
Top 10 similar courses to Arts and Heritage Management are:

Food & Beverage Management
International Law in Action: A Guide to the International Courts and Tribunals in The Hague
Fundamentals of Immunology: Innate Immunity and B-Cell Function
Introduction to International Criminal Law
Econometrics: Methods and Applications
Introduction to Neuroeconomics: How the Brain Makes Decisions
International Humanitarian Law in Theory and Practice
High Stakes Leadership: Leading in Times of Crisis
The Technology of Music Production
Corporate Sustainability. Understanding and Seizing the Strategic Opportunity
Moral Foundations of Politics
```

Figure 15: recommended courses using the first recommending system



```
Enter difficulty level of the course( Beginner, Intermediate,Conversant,Advanced): Beginner
Enter a list of keywords : arts heritage management

 Our top 10 most similar courses for the keywords ['arts', 'heritage', 'management'] are:
1 'Arts and Heritage Management' with a similarity score of 0.147
2 'What Is Contemporary Art?' with a similarity score of 0.086
3 'Management of Fashion and Luxury Companies' with a similarity score of 0.078
4 'Managing the Company of the Future' with a similarity score of 0.070
5 'Understanding Research Methods' with a similarity score of 0.069
6 'Global Environmental Management' with a similarity score of 0.049
7 'Introduction to Software Product Management' with a similarity score of 0.046
8 'Data Management for Clinical Research' with a similarity score of 0.041
9 'Intercultural Management' with a similarity score of 0.039
10 'Strategic Business Management - Microeconomics' with a similarity score of 0.037

 -----------------------AFTER FILTERING based on the given difficulty level-------------------------------------

 Our top 10 most similar courses for the keywords ['arts', 'heritage', 'management']  are:
0                       Arts and Heritage Management
1                 Managing the Company of the Future
2                     Understanding Research Methods
3        Introduction to Software Product Management
4                           Intercultural Management
5     Strategic Business Management - Microeconomics
Final result after post-filtering based on mean_compound feature :
 0                      Arts and Heritage Management
1                 Managing the Company of the Future
2                     Understanding Research Methods
3        Introduction to Software Product Management
4                           Intercultural Management
5     Strategic Business Management - Microeconomics
```

Figure 16: recommended courses using the second recommending system

References

[1] *course reviews on coursera.* https://www.kaggle.com/datasets/imuhammad/course-reviews-on-coursera?select=Coursera_reviews.csv.

[2] *coursera courses dataset.* https://www.kaggle.com/datasets/khusheekapoor/coursera-courses-dataset-2021.

[3] Anna Veronika Dorogush, Vasily Ershov, and Andrey Gulin. “CatBoost: gradient boosting with categorical features support”. In: *arXiv preprint arXiv:1810.11363* (2018).

[4] Michael J Pazzani and Daniel Billsus. “Content-based recommendation systems”. In: *The adaptive web*. Springer, 2007, pp. 325–341.

[5] *Silhouette Analysis.* https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html.

[6] Ramni Harbir Singh et al. “Movie recommendation system using cosine similarity and KNN”. In: *International Journal of Engineering and Advanced Technology* 9.5 (2020), pp. 556–559.

[7] *TF-IDF for machine learning.* https://www.capitalone.com/tech/machine-learning/understanding-tf-idf/?fbclid=IwAR3-wp9MOAPNfHskmGZ9cGxLh2UjdrIDQYkv2HJr5T76TFo8cslAziXJ

[8] Jiawei Yao. “Automated sentiment analysis of text data with NLTK”. In: *Journal of Physics: Conference Series*. Vol. 1187. 5. IOP Publishing. 2019, p. 052020.

Appendix

Parts	**Names**
Introduction	Ghada Ben Ammar and Hajer Ben Ammar
Dataset overview	Ghada Ben Ammar and Hajer Ben Ammar
Data pre-processing and feature engineering	Everyone
Data comprehension and analysis	Ghada Ben Ammar and Hajer Ben Ammar
Recommendation system implementation: First Method	Maysa Haddar
Recommendation system implementation: Second Method	Ghada Ben Ammar and Hajer Ben Ammar
Sentiment Analysis	Mayssa Haddar
IMDB score generation with two methods	Oussama Zaibi
Combined features	Oussama Zaibi

Table 2: Project Roles