

# A Recommendation System Involving a Hybrid Approach of Student Review and Rating for an Educational Video


Manar Joundy Hazar   [University of Al-Qadisiyah, Monastir University | [manar.joundy@qu.edu.iq](mailto:manar.joundy@qu.edu.iq)]

Salah Zrigui  [ Laboratoire d'Informatique de Grenoble | [salahzrigui@gmail.com](mailto:salahzrigui@gmail.com) ]

Mohsen Maraoui  [ Monastir University | [maraoi.mohsen@gmail.com](mailto:maraoi.mohsen@gmail.com) ]

Mounir Zrigui  [ Monastir University | [mounir.zrigui@fsm.rnu.tn](mailto:mounir.zrigui@fsm.rnu.tn) ]

Henri Nicolas  [ University of Bordeaux | [henri.nicolas@labri.fr5](mailto:henri.nicolas@labri.fr5) ]

 *Research Laboratory in Algebra, Numbers theory and Intelligent Systems (RLANTIS), Monastir University, Monastir 5019, Tunisia*

**Received:** 16 December 2022 • **Accepted:** 12 May 2023 • **Published:** 06 October 2023

**Abstract** Video recommendation systems in e-learning platforms are a specific type of recommendation system that uses algorithms to suggest educational videos to students based on their interests and preferences. Student's written feedback or reviews can provide more details about the educational video, including its strengths and weaknesses. In this paper, we build an education video recommender system based on learners' reviews. we use LDA topic model on textual data extracted from educational videos to train language models as an input to supervised CNN model. Additionally, we used latent factor model to extract the educational videos' features and learner preferences from learners' historical data (ratings and reviews) as an output CNN model. In our proposed technique, we use hybrid user ratings and reviews to tackle sparsity and cold start problem in the recommender system. Our recommender uses user review to suggest new recommended videos, but in case there is no review (empty cell in matrix factorization) or unclear comment then we will take user rating on that educational video. We worked on real-world big and diverse learning courses and video content datasets from Coursera. Results show that new prediction ratings from learners' reviews can be used to make good new recommendations about videos that have not been previously seen and reduce cold start and sparsity problem effects.

**Keywords:** Educational video, Recommendation systems, Sentiment analysis

## 1 Introduction

Online learning platforms are learning methods that use modern technologies like the Internet which allow offering courses to students anytime from anywhere and interact with students efficiently and straightforwardly. To keep up with the rapid development of educational institutions worldwide, e-learning has joined the essential requirements of the educational process Hazar *et al.* [2019], Legrand *et al.* [2019]. Among the most often-used machine learning applications on modern websites and platforms is recommendation systems. Due to the rapid growth of e-commerce, personal recommendations to customers are necessary for an e-store to stand out. Recommendation systems have constituted the foundation of online services like Spotify, Netflix, YouTube, and Amazon.com Ricci *et al.* [2021]. The recommender system (RS) is a new big data technology. Recommendations in the form of user-friendly guidance help students succeed in an online learning environment. Many of the recommendation system's strategies use both collaborative and content-based filtering. Both of these methods may be used for online learning, and additional methods like hybrid, information-based, and so forth Afoudi *et al.* [2021], Kulkarni *et al.* [2020]. The educational system is improving with the aid of the Internet and e-learning. The video material helps the user learn easily or to teach quickly. Therefore, a recommendation system with user reviews or ratings based on video material in the e-learning platform can accomplish this

task effectively Ali *et al.* [2022]. The rise in online video consumption has led to a new problem: information overload. This required developing an automatic video recommendation system to provide learners with relevant videos that match their requirements Dias *et al.* [2021]. As a result, many academic and commercial institutions have focused on developing video RSs. YouTube, Yahoo! Video, and Bing Videos are commercial sites that use deferent information formats to build their RSs, frequently relying on textual video data and registered user profiles. Generally, good video RSs should be built on analysing users' interests based on their usage history or profiles Yassine *et al.* [2021]. Since 2020 (during the COVID-19 epidemic), more people have signed up for popular online learning platforms, including Coursera<sup>1</sup>, Khan Academy<sup>2</sup>, lecture video channels on YouTube<sup>3</sup>, and various university platforms like Stanford and MIT Chen *et al.* [2022], Slimi *et al.* [2020]. In this research, we used information from user reviews and comments to address one of the problems RSs frequently experience. RSs use a single discrete scale to model customers' thoughts about an item, which is not entirely accurate for expressing the user's preferences. In section 2 we give a background about all subjects used in this article, section 3 is a related work, section 4 explain the proposed approach, section 5 simulation setting, experiments and results presented in section 7 and section 7.4 is a rating hybridization of our model.

## 2 Background

### 2.1 Video Recommendation Systems on an E-learning Platform

User can recommended videos using a video recommendation engine that analyzes videos they have seen or are currently watching. Besides saving customers from having to browse through many movies to find their favourites, this technique increases network traffic and user stickiness on video websites. The most crucial component of a video recommendation system is the video recommendation algorithm, which primarily comprises two filtering types: collaborative and content-based Xiao *et al.* [2018], Kundu *et al.* [2021]. In handling information overload, recommendation systems have proven successful. the exponential growth of data and material that is available online. Every day, billions of web pages, social media postings, videos, photos, and other digital content are created and shared on the Internet, making it an enormous reservoir of information.. Users were frequently presented with abundant products and e-learning resources. Therefore, personalization is a key tactic for enhancing the user experience. These systems have proven essential in different web domains, including e-learning websites El Moustamid *et al.* [2017], Tolety and Prasad [2022]. Aided by the Internet and e-learning, the educational system has improved. So the video material helps whether student to learn the subject easily or the teachers to teach the subject easily. Therefore, the recommendation system with user reviews or user ratings based on the video material on the e-learning platform can accomplish this task effectively Phongsatha and Cleesuntorn [2017], Wong *et al.* [2022].

### 2.2 Recommendation System-based User Rating

El Mabrouk et al. El Mabrouk *et al.* [2017] proposed a hybrid intelligent recommendation system for e-learning systems based on data mining. The recommendation system's objective is to direct users of e-learning platforms toward the most important content. Making material easier to obtain helps people develop their centre of interest. The system consists of four module, the first module is for implicit and explicit data collecting. The second module is used to process the data gathered and construct the learner profile, user categorization, and content classification. The fourth module creates log files and user history data for use in the forthcoming recommendation processes, while the third module generates recommendations based on the suggested model.

### 2.3 Recommendation System–Based User Reviews

Islek and oguducu Islek and Oguducu [2022] recommendation systems regarded as essential elements of the e-commerce industry owing to their direct financial influence. Items considered relevant to the client discovered by utilizing their prior activities. These items were presented appealingly, thanks to recommendation algorithms. By considering consumers' most recent purchase history, Islek and Oguducu

described the architecture of the recommendation system. They also suggested a hierarchical item recommendation system for the e-commerce industry. Two layers comprise the suggested hierarchical structure. The recommendation level is the second level of the first level of the recommendation system, which has three phases. In this system, the input to the recommendation model was enhanced using cutting-edge embedded learning techniques. Creating a document for each item based on its text is the first level's first stage. Several text-preprocessing procedures were performed on these documents in this level's second phase. Each item document is transformed into a fixed length-embedding vector in the third stage, item embedding construction. In the second level of the suggested hierarchical system, the last component is a self-attentive sequential recommendation model. Instead of an item id, each item at this level of the self-attentive network was represented by a fixed-length embedding vector. They use the textual data associated with each item to produce a document for each in the first phase of the hierarchical system's first level. Every e-commerce platform has an item detail page to provide comprehensive information. Typically, this page includes pictures of the item and some text, like the title, description, and reviews. A buyer views an item's detail page to decide whether to buy it.

## 3 Related Work

Murad et al. Murad *et al.* [2018] examined the RS papers produced by academics in the domains of computer science, information systems, and information technology. The data mining, sentiment analysis, learning analytics, and application RS on large data is the study that discovered problems in RS domain research. Herath and Jayaratne Herath and Jayaratne [2017] suggested that we must motivate e-learners to actively engage in an e-learning environment to improve education. This system uses web mining techniques, particularly web usage mining, to identify the navigational preferences of e-learners and web content mining to find pertinent online materials. The subsequent parts of the web mining process include data cleaning, integration, transformation, and pattern recognition. The first step is data purification, a part of pre-processing. The server's access logs should be cleaned up in this phase to remove unnecessary and duplicated references. Data integration is the subsequent phase. This is because not all of the information from web mining can be found in the server's access logs. Different e-learners must be linked to their sessions or transactions, online content data, profiles, results data, assigned task data, and other relevant data stored in the database throughout this stage. Thakker et al. Thakker *et al.* [2021] presented in the research movie recommender systems. Websites must provide individualized services to each user to increase customer happiness and the quality of the customer's time engagement when a user has a wide range of service options. Users are often known for being unable to choose; thus, when several options are available, they choose nothing or choose poorly. According to a poll, a typical Netflix or other movie-streaming customer loses interest after 60 to 90 seconds of perusing up to 20 films. If this occurs, it generates consumer unhappiness, ultimately

causing the loss of the client. Therefore, it is crucial to create an effective recommender system that greatly aids users in making decisions. These systems have been created utilizing unique methods, and applying such approaches has generally helped e-portals grow their companies. Suggestions made by the site increase the likelihood that users will believe them. They showed the user interfaces of two widely used streaming services—Netflix and Amazon Prime Video—and how they present the outcomes of suggestions made specifically for the user. Kim et al. Kim *et al.* [2021] suggested a technique for producing a new suggestion candidate for users by using a learning model that anticipates ratings across several categories and total ratings from reviews. Through word embedding in the user review data, CNN-BiLSTM was used to predict the user’s rating for each criterion, and a linear regression model was used to predict the user’s total rating. We determined and applied the user and item priority of the criteria during this step. The weights taken from the linear regression model are used for the user’s priority of criteria, while the average value is generated by averaging the ratings for each criterion based on the item used for the item’s priority of criteria. The anticipated overall rating of the user’s item was then synthesized to create a recommendation candidate. Using the Tripadvisor dataset, the suggested approach was applied to the user’s hotel suggestions. The experiment verified the suggested model’s excellent performance.

## 4 Proposed Approach

We proposed a recommender system based on the educational videos’ content and learner reviews to satisfy what students actually need to learn. Our model builds on a combined content-based recommendation system and convolutional neural network (CNN) that runs on three levels, as illustrated in figure 1.

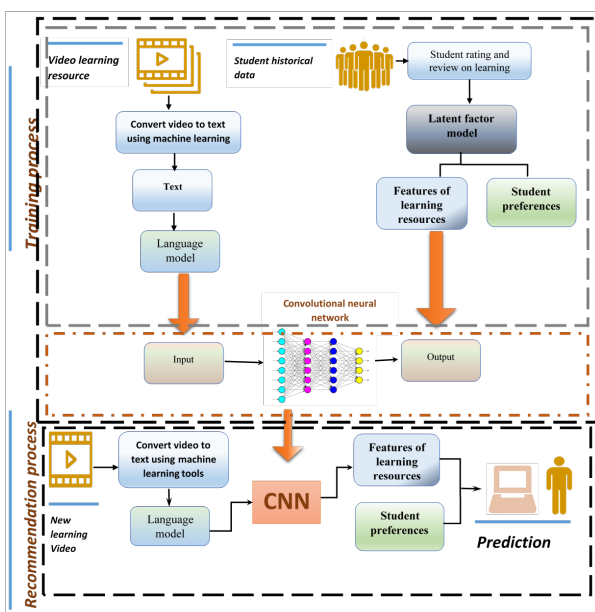


Figure 1. Model of Educational video recommendation system

Level one in figure 1 on the left line is an input of a CNN training model. We used a language model of educational

videos after preprocessing phrase of source videos. Where the right line is an output of the CNN training model, we use the latent factor model (LFM) for learners’ historical data. Learners’ reviews of educational videos were used as training data for LFM. At this level, we made a preprocessing step to formulate the CNN model’s input and output data. Level two is a training level at which all CNN parameters will be trained by the input and output data. Level 3 is the recommendation level. At this level, a new educational video will be recommended.

### 4.1 Preprocessing

To build a language model (input of the CNN model), we must train it on textual data. Thus, we preprocessed source educational videos to convert them to text data. The first step was extracting audio from educational videos; second, we exploited Google’s API to extract textual data from audio, keeping track of the original contents. This transformation was done automatically by speech-to-text operation (see figure 2. API borrowed from Google had an audio length limitation, which was a challenge for our model, so we fixed it with our source code, breaking down long audio sequences into small junk. Then, we combined text from all junk belonging to the same source video. This process keeps the original content of the source video stable, as we proved in section 7.1.1.

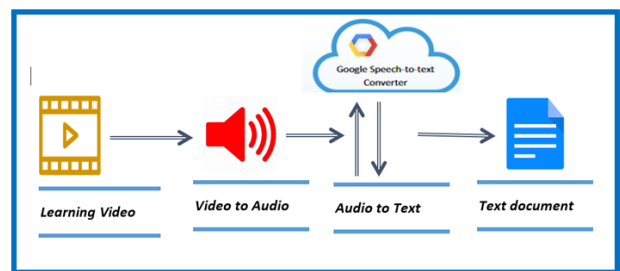


Figure 2. Model of Educational video recommendation system

### 4.2 CNN Model

The language model will provide the input training data for the CNN model. We used a language model on textual data extracted from the source video. LFM will be the output of CNN, trained from learners’ histories represented by free text reviews on educational videos. The user can express their opinion and preference by writing data more than numerical data (rating data), which abbreviates what he thinks in one number. For instance, a user may rate a particular item highly but have a negative opinion on a small part of the video, like the teacher’s teaching style. Those details would be invisible for numerical data but are very important, especially for the RS. The CNN model built on four layers: The first layer is a convolutional layer with multiple feature maps. The second layer is a pooling layer, and in this layer, we apply a max-pooling function to reduce dimension. The third layer is a convolutional layer, and the last layer is a fully connected layer. Figure 3 shows the CNN model.

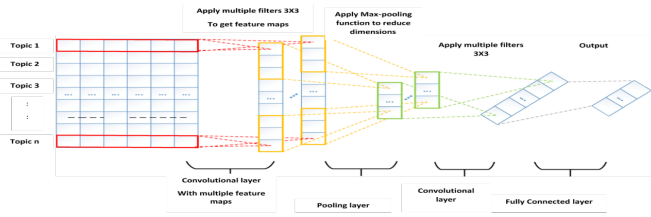


Figure 3. CNN model

### 4.2.1 Language Model

A statistical model that predicts the probability distribution of word sequences in a given language is known as a language model. It is trained on a huge text corpus and utilized in natural language processing applications such as speech recognition, machine translation, and text synthesis. A language model’s purpose is to predict the likelihood of a specific sequence of words happening in the language based on the probability distribution of words observed in training data Wei *et al.* [2023]. The language model was used to justify the input of CNN. LDA or Latent Dirichlet Allocation is a probabilistic topic modeling technique for discovering hidden topics in the text documents. It is assume that each document is a collection of different topics, and that each topic is a distribution over a set of words. The system estimates the probability distributions of topics and words to infer the latent topic structure of the corpus, and then assigns each word in each document to a specific topic based on these probabilities. The topic model that emerges can be applied to applications such as document classification, information retrieval, and recommendation systems. Because LDA enables us to acquire a topic from each educational video, we can use its output as a feature vector for educational videos to configure items according to a content-based recommendation system. After we convert the source video to its textual data, each video represented by a document. A document contains latent topics. Every single topic represented by a probabilistic distribution of words. LDA assists with extracting topics related to that document; it scans the document and discovers the statistics of the words, then represents those words as a probability of topics Ayadi *et al.* [2014]. Figure 4 shows the educational video recommendation model.

### 4.2.2 Latent factor model (LFM)

LFM use to justify the output of the CNN model. LFM extracts the educational videos’ features and learner preference from historical data; these data represented in two matrices: U refers to user preference, and V refers to features of educational videos. With LFM, we can predict the user rating on any educational video based on some indirect factors and detect the learners’ interests. Figure 5 shows LFM for educational videos and learners’ reviews. Tradition LFM reduces solution space by L2-norm regularization. That cased in over-smoothing problem. Thus, in our propped model, we use LFM by L1-norm regularization to indicate educational resource features and learners’ preferences. Matrix factorization modified according to equation 1 below

$$j(U, V) = \sum_{ij} (U_{i*} - V_{*j} - r_{ij})^2 + \lambda_1 |U|_1 + \lambda_2 |V|_{1,2} \quad (1)$$

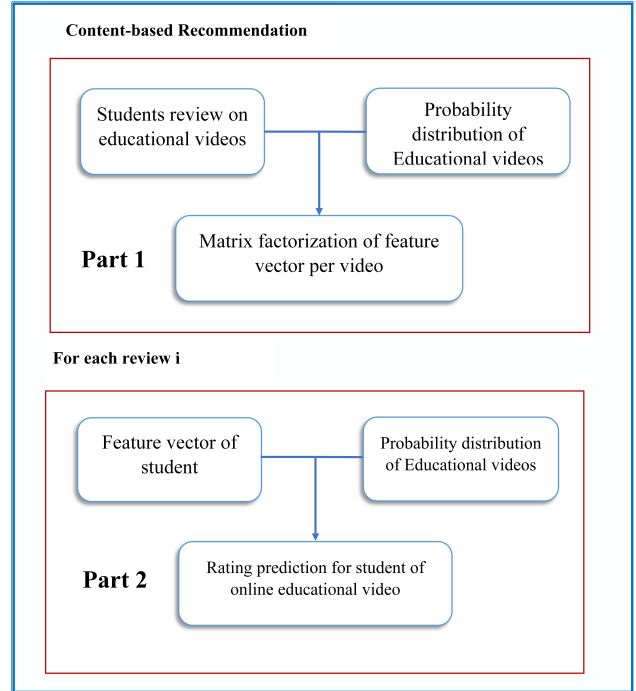


Figure 4. Educational video recommendation model

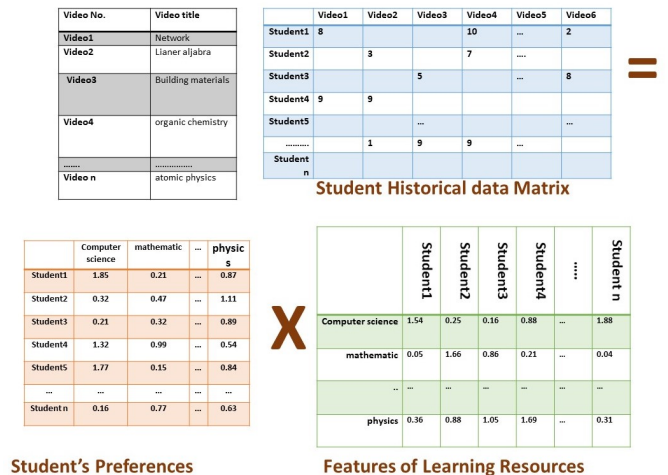


Figure 5. Process of the Latent factor model (LFM)

$j(U, V)$  is the fidelity part, and the rest of equation 1 is the regularization part.

U: It is a matrix of relations between learners and latent factors.

V: It is a matrix of relations between educational videos and latent factors.

$r_{ij}$ : It is a rating given by learner i on educational video j.

$\lambda_1, \lambda_2$ : are the regularization parameters to balance the fidelity and regularization parts.

As we mentioned previously, we applied these computations on non-empty cells in the matrix

to minimize the non-convex function in the equation, cross-validation and a grid search were used to optimize parameters. A cross-validation procedure runs for the proposed recommendation algorithm, and then the LFM model captures the mathematical framework.

## 5 Simulation Setting

We run our proposed model on the configuration of Python 3.8.8 on the Anaconda environment and with a spider. A personal computer had the following description: HP Elite x2 1012 G1, Processor: Intel(R) Core(TM) m7-6Y75 CPU @ 1.20GHz 1.51 GHz, Installed RAM: 8.00 GB (7.88 GB usable). The system type is a 64-bit operating system with an x64-based processor. For further video processing and saving, we need to rent a server (GPU DEDICATED SERVER) from interserver.net<sup>1</sup> Company with the following description in figure 6.







| SINGLE GPU   |  |
|--|--|
| <br>AMD Ryzen 3900x | <br>64GB Memory                       |
| <br>2 x 1TB NVMe    | <br>1 x NVIDIA GeForce RTX 2080ti GPU |
| <br>150TB Transfer | <br>1GB Port/30 IP Space             |

Figure 6. GPU server configuration

## 6 Dataset

We worked on the same dataset in the previous two versions of this article, Hazar *et al.* [2022a], Hazar *et al.* [2022b], which were from the Coursera platform; the data size was 1.45 million comments and reviews written by learners on educational videos. The data are described in table 1. part 1 contains all data about 622 learning courses, while part 2 contains all data on 1.45 million learner scorings and comments.

For testing purposes, we reduced the data size and selected three courses from the dataset. Those three courses contain 178 educational videos, 198856 learners' comments, and the same number of learner ratings from 1 to 5 stars on source videos. The three courses used can be accessed via the Coursera platform as follows:

1. machine-learning course from Stanford University URL: <https://www.coursera.org/learn/machine-learning>
2. python-data from University of Michigan URL: <https://www.coursera.org/learn/python-data>
3. e-learning from University of Illinois at Urbana-Champaign URL: <https://www.coursera.org/learn/elearning>

These courses have been chosen for many reasons, such as the courses' experts, authority and quality of spoken English.

<sup>1</sup><https://www.interserver.net/>

Table 1. Disruption of dataset

| Course Description         |           |  |
|----------------------------|-----------|--|
| Variable                   | Class     | Description  |
| Name                       | Character | Course Name  |
| Institution                | Character | Reviewer Name  |
| course_url                 | Character | Course URL   |
| course_id                  | Character | Course ID  |
| Student Review Description |           |  |
| variable                   | Class     | description  |
| reviews                    | character | course review  |
| reviewers                  | character | the name of reviewer who wrote the review            |
| date_reviews               | date      | date the review was posted                           |
| rating                     | integer   | the rating score given to the course by the reviewer |
| course_id                  | character | course ID  |

All data used to train and test our proposed model were saved in two CSV files. The first file contains learner reviews (198856 reviews) and their original ratings (from 1 to 5 stars) on the 178 educational videos in the three selected courses as shown in table2. The second file contains details and information related to the educational course.

Table 2. CSV File of Learners' Reviews and Ratings

| reviews  | reviewers   | date_reviews | rating | course_id   |
|--|-------------|--------------|--------|-------------|
| Dr. Chuck is the best mentor I've ever been taught by. This course helped me a lot and surely will help me afterwards also <3 thank you COURSERA | By Dhiraj J | 28-May-20    | 5      | python-data |

## 7 Experiments and Results

### 7.1 Prepressing

In this initial step, educational videos are converted to textual data, as mentioned in section 4.1, figure 2, by a sequence of converting operations: video to audio, then audio to text by speech-to-text Google API. To deal with API restrictions on the audio length, we divide long audio into small chunks based on the lengths of the teacher's silences in the video, creating split points when a speaking person stops for five seconds. After that, all split texts for the specific video collected into one document representing all textual data of that video without any information loss. That needs to be approved; hence, we compare the final collected text file resulting from this step with the transcript text file provided by the Coursera site on the web page below each video. As an example, figure 8 shows the converted text, and figure 7 shows the original text from the Coursera site for the educational video titled "Lists" from the "python-data" course accessed by the link <https://www.coursera.org/learn/python-data/lecture/ss3c1/8-1-lists> on 8 April 2023.

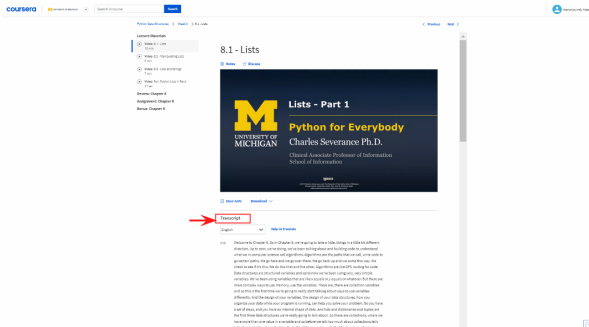


Figure 7. video and transcript text file on Coursera website

```
welcome chapter 87 chapter 8 things in a little bit different direction up to now we've been talking about building code. To understand what we do
computer science call algorithms algorithms are the parts that we wrote right code to go here and we go over there we go back up and we
come this way. Check to see if it's this we do this the algorithms are like: Go, floating for code. The structures are: structured variables using very
very simple variables we've been using variables that are like x equals or y equals or whatever but there are more complex ways to use memory use the
variables there over there are collection variables so. This is the thing that we're going to start talking about ways to use variables
differently. And the design of your variables the design of your data structures. Now you organize your data while your program is running can help you
solve your problem. So you have a set of steps here an internal these nodes. And lists and dictionaries in complex use the first three days of structures
were really going to talk about. So these are collections where we have. More than one value in a variable and so before we talk too much about collections
let's talk about what's not a collection. Example 8 equals to that. Find out in this world, we'd be actually been using lists all along. When we see
equals 4 it finds that same piece of memory overrides the two with a four. So when it's all said and done what we have is a 4. Now, a list is a collection
meaning that in a single variable we can have more than one thing. And so in this world, we'd be actually been using lists all along. When we see
square brackets in that we've been using these in for statements exceptors exceptors etcetera. This is a list. Square brackets in a list constant. But what's
your meaning here is a list. These things these different things. Joseph plans and calls. Into a single variable by enclosing them in those brackets.
And so it's still a single variable. Friends is no longer a string it's a list of strings which is different than a string. And the same for carry on one
point and surface. The set kind of it is a sort of like a list of things with the same 'o' or 'j'. But then you can put multiple things in it and the
interesting thing about these data structures that will use for different purposes. Is how you put them in and then how you get them back out will be
the distinction between these. Three different kinds of collections that we're going to talk about. So like I said list constants are anything in square
brackets so that's a list of. Three integers 124 + 76. This is a list of three strings red yellow and blue you see it can be either kind we just use commas
and we use square brackets to start them and stop them. And you can have an empty listing so that is. List constants are those are square brackets. I like I said we have
integer 24 and then 98 is union is a floating point. And it doesn't even have to be a we won't do this too much for a while but this is a three element list.
The first element in the list is an integer the second element in the list is another list. It's a list which contains all the way down this description
so you can have nested lists and. Talk about data structures in the more advanced, make a list that includes another list. In just part of the
design out. We're all way premature to talk about why you would do this but it is possible. This is a one a list and a seven and so there are actually three
elements in this particular list. x, and you can have an empty listing so that is. List constants are those are square brackets. I like I said we have
been using this already. We have four loops and we've iterate the eye through 343 + 2 and 121 parentheses. Five things which means at it is going to run
this once. For each value in the list and I was going to use on the successive values in the list and like I said we have already been doing that right and
we've been doing that. Since it's like that. Solve and definite loops the kinds of things that we've been doing all along our best friends the end so you
can think of a list as like a set. The set of three strings and are like for all the members of this set. And so for friends in friends, and remember the
fact that I use plural here and singular here does not mean anything to python it means something to you as your reading the code that's a mnemonic
variable. Can be anything you want, articles this is exactly same code. Friends Joseph plans blue and poor size on takes on the successive value, python
sees these two things. As completely equivalent. But you as a human being read them as something different and so that's. The choice of reminding us of
the choice of variable name. So these collections they're putting stuff in the collection and getting more things in the collection but there's also getting
them out, and shall we use the index operator the same operative we use to look inside strings to look inside list so I read this in my mind as sub-
this is friends sub-1. Remember that. Here, using the elevator scheme in source technique. The first one is zero the second one is one and so friend
someone is actually the second item in the list and that's how gloom ends up getting printed out here. So you can look inside list. We can change listen the
word that's used in python speak is called mutable means changeable. I'm so you'll notice that some things are not mutable. So like strings so we put in
banana. And fruit this seems like it would change that first letter to lowercase b we got sub-zero fruit sub-zero equals b. I thought you saw this trace
back. You know you might think this is no problem syntactically it's python would actually support it but it doesn't because we're not allowed to change the
string and that's part of the reason that. When we do things like lower on strings we always say it makes a copy of the string and gives us back a
lowercase copy without changing the original string and that's. Because strings are not mutable. And that's what we do when we use python uses to talk about
```

Figure 8. Converted text file

### 7.1.1 Originality Source Data Evaluation

In this experiment, we try to prove that our source input data preprocessing preserves information originality and accuracy. Therefore, we calculate the distance (similarity) between textual data after the conversion process and the original transcript text provided by Coursera below every educational video in the dataset. We apply two strategies of similarity. First, we use the Python and Cython-written natural language processing module SpaCy library. Researchers in Treude *et al.* [2015] demonstrated that the SpaCy library is significantly faster than many other libraries. Then, we have included cosine similarity because it is a simple, reliable similarity metric used with the TF/IDF weighting scheme Mahmoud and Zrigui [2021]. Since the videos in our dataset for the three courses differ in size, length, and type of material (science, math, etc.), we calculate the similarity of different video samples from the source dataset. Since the data set is so large, table 3 show only a very small sample for documentation purposes, which is used for result display and discussion.

In figure 9, we use the same small sample as table 3 to explain and present the results because the dataset is too large to exhibit here. Figure 9 illustrates the great similarity between the two texts; some brief especially non-mathematical videos has a similarity index of 100%. Therefore, we can observe the following: 1- In all 10 shown educational videos, spacy is a built-in module in the version of Python that we used to create our model. There is no information loss throughout the computation process, and as a result, spacy provides higher similarity than cosine. 2- Some videos performed more similarly than others in both metrics, fully driven by the video's content. Where the videos contain a lot of mathematical information, this kind of information not always completely converted to text 100%. since, for instance, symbols may be translated into words or formulas translated

Table 3. Details about small sample of educational videos from dataset

| Video ID | Title   | Length (min.) | Size (MB) | Course ID        |
|----------|---|---------------|-----------|------------------|
| Vd. 1    | Manipulating Lists  | 9:13          | 12.6      | Python-ata       |
| Vd. 2    | Processing Files  | 11:42         | 40.0      | Python-ata       |
| Vd. 3    | Counting with Dictionaries  | 11:52         | 14.7      | Python-ata       |
| Vd. 4    | Tuples  | 17            | 20.0      | Python-ata       |
| Vd. 5    | Active Knowledge Making, Part 2C: Memory Work in Learning                 | 5:13          | 8.67      | e-learning       |
| Vd. 6    | Recursive Feedback, Part 4B: Summative Assessment vs Formative Assessment | 10:38         | 16.2      | e-learning       |
| Vd. 7    | Metacognition, Part 6B: Metacognition in e-Learning Ecologies             | 11:40         | 17        | e-learning       |
| Vd. 8    | Applications of Machine Learning  | 4:28          | 9.51      | Machine-learning |
| Vd. 9    | Gradient Descent for Multiple Linear Regression                           | 7:45          | 7.23      | Machine-learning |
| Vd.10    | Logistic Regression   | 9:48          | 8.60      | Machine-learning |

into sentences. Since all evaluated videos had an index of similarity greater than 95%, the overall result appears satisfactory.

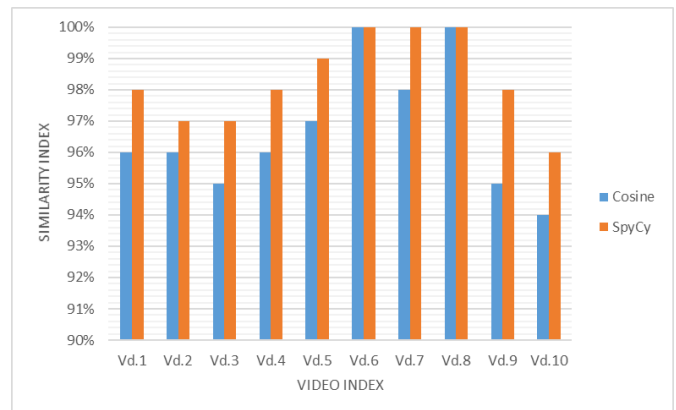


Figure 9. Similarity measurement

### 7.2 Leaner historical data preparing

User historical ratings are a common basic input source for a recommendation system. These data are numerical, which seems restricted in expressing user opinions and interests. Therefore, free text written by the user will be good data to extract his/her preference to build an accurate RS. This is why we build our educational video recommendation system based on learners' reviews through sentiment analysis of their opinion to extract preferences from all information in written text comments on learning resources to match their learning needs. In this study, we develop our learner comment-based RS model for a variety of reasons Zhang *et al.* [2015] that are briefly discussed in the following items:

1. We can use learners' comments and reviews to deal with sparsity issues, where text comments offer helpful information in the absence of a rating that can be used to predict a rating.
2. Free review text is a good choice to fix cold-start problems by providing all necessary information about new

items or new users.

3. Learners’ reviews are the best choice for dense data problems, causing it to deliver more features to make an accurate recommendation system performance by guaranteeing high-quality ratings in case of conflict between ratings and comments of the same user on the same source; for example, user x rates item y 5 stars incorrectly, and the same user x on the same item y writes a very negative comment.
4. User reviews powered in fine-grained sentiment can help in extracting more features about a single item that will be very flexible to personalized use/item models.

We use sentiment analysis to classify user comments as positive, natural, and negative. For this purpose, we apply natural language processing tools. VADER (Valence Aware Dictionary for Sentiment Reasoning) is one of the most suitable sentiment analyzers Hutto and Gilbert [2014]. VADER is a lexicon employed to make a sentiment analysis of textual data in different domains, such as social media, education, and news. It is a rule-based sentiment analysis tool.

### 7.2.1 Review Sentiment Analysis

We analyze a sample of 198,856 negative and positive learners’ reviews on 178 educational videos from three different courses. This sample is handpick from the bigger real-world dataset of 1.45 million reviews on 622 courses presented by the Coursera online platform. The NLP toolkit of sentiment analyzer VADER is use to classify and score the reviews based on English adjectives in learner comments. Additionally, VADER’s lexicon allows qualitative analysis that can strongly detect all characteristics and proprieties of text review that influence how strongly the text’s sentiment is perceive. Then we have five syntactical and grammatical heuristics that can modify and change the intensity of review sentiments as well, exemplified and highlighted in red in table4. These heuristics take into account term relationships that are sensitive to word order. The five heuristics listed are as follows:

1. Exclamation point (!) punctuation, which keeps semantic orientation stable but increases the intensity score. For example, the learner comment “this course is good!!” will be a more intense comment than “this course is good”.
2. Full capitalization, or ALL CAPS, which also keeps semantic orientation stable and will highlight the sentiment of the relevant term. That increases the intensity score of the sentiment. For example, “this course is GOOD” will be more intense than the comment “this course is good”.
3. Adverb degree. Some words intensify the sentiment intensity or reduce intensity. For example, the comment “this course is extremely good” will be more intense than “this course is good.”
4. “But” conjunction, which affects shifting sentiment scoring; the text that comes after “but” will be the main learner opinion. In this case, the comment contained a mixed sentiment, such as “lecture is great, but need

more explanation,” and the second part of the comment identified the sentiment polarity.

5. Most words can influence sentiment polarity, and flipped text scoring is a negation “not”. For example, consider the comment “This course is not good.”

### 7.2.2 Rule-Based Sentiment Analysis

A rule-based sentiment analysis or sentiment lexicon consists of a set of rules called a lexicon which can classify words as negative, positive, and natural according to intensity Pandey [2018]. To apply a rule-based sentiment analysis, we need to perform the following steps on learners’ text reviews:

- Data extraction.
- Text tokenization, splitting text into separate words.
- It removed stop words, such as “a”, “an”, “the”, “they”, etc., that were not used in the sentiment analysis.
- Some of the punctuation not used in the sentiment analysis should be removed.
- Text preprocessing based on sentiment lexicon to detect the polarity of emotion.

**Manual Example:** For more explanation and clarification, let us take the learner’s comment, “This course is good,” to apply the previous rule-based sentiment analysis steps as shown in figure 10.

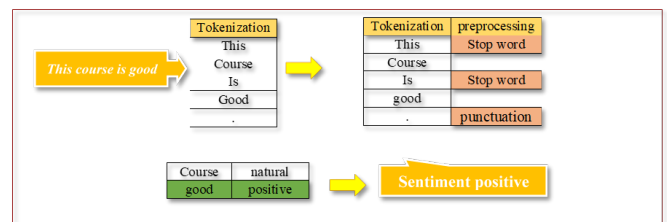


Figure 10. Rule-based sentiment analysis

### 7.3 Rating Prediction

Sentiment polarity was employed to rate the learner’s comment. The text review already had been classified by the VADER lexicon as positive, negative, or natural. To detect the polarity sentiment in an individual review, a compound value was used. We used (1) as a threshold of compound value to classify learners’ comments where:

- If the compound value > 0.001, then the sentiment was positive, and polarity = 1.
- If the compound value > -0.001 and < 0.001, then the sentiment was natural, and polarity = 0.
- If the compound value < -0.001, then the sentiment was negative, and polarity = -1.

Table 5 shows the CSV file’s details resulting from text review classification and rating.

VADER lexicon scores are between +4 and -4, where +4 is a very positive sentiment and -4 is a very negative sentiment. In our proposed framework, we put polarity between +1 and -1. For normalization and comparative purposes, we need

**Table 4.** Syntactical and Grammatical Heuristics of Sentiment Analysis

| Review   | Reviewer Name  | Date of Review | Rating | Course           | URL of Review Access Webpage  |
|--|----------------|----------------|--------|------------------|---|
| Happy to learn and feeling motivated to explore more!!                       | By Charu A     | 11-Jul-2020    | 5      | python-data      | <a href="https://www.coursera.org/learn/python-data/reviews?star=5&amp;page=93">https://www.coursera.org/learn/python-data/reviews?star=5&amp;page=93</a>   |
| <b>GREAT COURSE</b> , very detail and great discussion board                 | By Gary C      | 2-Oct-2018     | 5      | python-data      | <a href="https://www.coursera.org/learn/python-data/reviews?completers=true&amp;sort=helpful&amp;star=5&amp;page=252">https://www.coursera.org/learn/python-data/reviews?completers=true&amp;sort=helpful&amp;star=5&amp;page=252</a> |
| Though-provoking and <b>extremely</b> useful to anyone working in education! | By Athina      | 30-Dec-020     | 5      | e-learning       | <a href="https://www.coursera.org/learn/elearning/reviews?star=5&amp;page=3">https://www.coursera.org/learn/elearning/reviews?star=5&amp;page=3</a>   |
| <b>not</b> useful  | By Dr S        | 18-Apr-20      | 1      | e-learning       | <a href="https://www.coursera.org/learn/elearning/reviews?page=1&amp;star=1">https://www.coursera.org/learn/elearning/reviews?page=1&amp;star=1</a>   |
| Lacks coding , <b>but</b> its very nice in terms of theory                   | By Francisco S | 30-Sep-2022    | 3      | machine-learning | <a href="https://www.coursera.org/learn/machine-learning/reviews?star=4&amp;page=3">https://www.coursera.org/learn/machine-learning/reviews?star=4&amp;page=3</a>   |

**Table 5.** Header information of CSV file from new rating prediction

| Text review         | Reviewer name                         | Review date     | User rating     | Course ID           | Sentiment type                 | Predicted rating |
|---------------------|---------------------------------------|-----------------|-----------------|---------------------|--------------------------------|------------------|
| Learner text review | Name of learner who wrote the comment | Date of comment | Original rating | Title of the course | Positive, negative, or natural | New rating       |

to formulate scoring fraction values between +1 and -1 to integer numbers from 1 to 5, as shown in figure 11. This value will represent a new rating protected from text-review sentiment analysis.

```

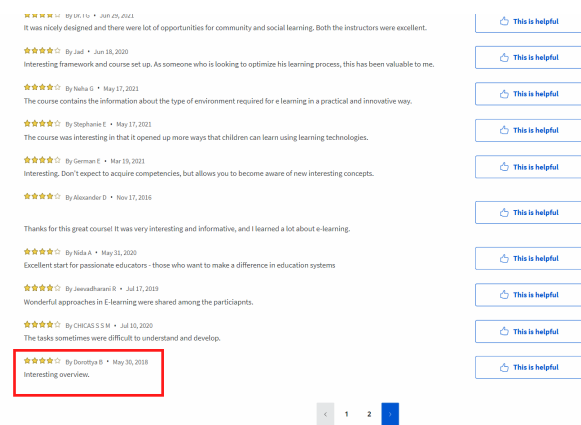
if vader_compound_score >= -1 and vader_compound_score < -0.5:
    Return 1
elif vader_compound_score >= -0.5 and vader_compound_score < -0.05:
    Return 2
elif vader_compound_score >= -0.05 and vader_compound_score <= 0.05:
    Return 3
elif vader_compound_score > 0.05 and vader_compound_score < 0.5:
    Return 4
else:
    Return 5
    
```

**Figure 11.** Vader compound value formulation

### 7.4 Rating Hybridization

Cold starts are among the most challenging aspects of any recommender system. Hence, in our proposed model, we try to solve it or at least reduce its effects on recommender system performance. We use a hybrid approach incorporating users’ historical data: rating and text review data. From text review data, we predict a new matrix rating. As previously mentioned, we analyzed learners’ reviews by sentiment lexicon to predict the user’s new matrix rating. However, not all users will comment on educational videos, which means if a user rates an educational video, they are not necessarily giving feedback; they may have provided a blank or non-readable comment that yields nothing or a wrong rating prediction. Thus, we will have empty cells in the new predicted matrix. In the original matrix, we also have empty cells and maybe the wrong rating value. For this reason,

we make a hybrid matrix factorization of the original and predicted ratings in case one of these reasons happen. In the new prediction rating, we will substitute it with the original rating in an equivalent cell in the hybrid matrix; otherwise, we will keep using the new rating. In table 3, we present different samples of the source datasets pre-described in section 6. Samples were selected to clarify the deference and accuracy between the original and predicted ratings. We carefully chose rating pairs with a high deference rating in the resulting matrix. In examining table 6, we see four deferent cases of rating degradation between the user and new predicted ratings that are fixed in the new proposed model; all four cases will be discussed in section 8. For more research reliability and transparency, figure 12 and figure 13 presented a real sample of learner reviews, and feedback in table 6 was captured directly from the Coursera site. In figure 12, we captured the user comment in row



**Figure 12.** captured real sample of learner comment and feedback on E-learning course

14 of table 6, ”Interesting overview.” This was accessed through the link ’<https://www.coursera.org/learn/elearning/reviews?page=1&star=4>’, which clearly shows that user ratings and our model’s rating predictions



**Table 6.** Selective sample of rating prediction from users' reviews using sentiment analysis

| Review -id | Text-reviews   | Reviewer name    | Review date | User rating | course_id        | Predicted rating |
|------------|--|------------------|-------------|-------------|------------------|------------------|
| 1.         | One of the best courses i studied in Coursera  | Baccouche M S    | 25-Feb-19   | 5           | machine-learning | 5                |
| 2.         | good   | SRUTHI P N       | 4-Jun-20    | 1           | machine-learning | 4                |
| 3.         | A bit boring!  | Abu M R          | 15-May-20   | 4           | machine-learning | 2                |
| 4.         | great course   | onkar p          | 3-Apr-18    | 4           | machine-learning | 5                |
| 5.         | Excellent...!  | RAHUL S          | 16-May-19   | 5           | machine-learning | 5                |
| 6.         | Nice   | Rupak K p        | 10-Apr-20   | 1           | python-data      | 4                |
| 7.         | best course  | rajyagurusaisrik | 3-May-20    | 2           | python-data      | 5                |
| 8.         | Good Lectures  | Himanshu V       | 12-Sep-20   | 3           | python-data      | 4                |
| 9.         | Great Course for starting!   | Mahmud M         | 23-Jul-18   | 4           | python-data      | 5                |
| 10.        | very good course actually!   | Dipu             | 26-Sep-16   | 4           | python-data      | 5                |
| 11         | It's good learning course  | Yogesh u         | 2-Aug-20    | 4           | python-data      | 4                |
| 12         | I really loved it.   | Szabolcs S       | 12-Mar-20   | 5           | python-data      | 5                |
| 13.        | Dr. Chuck is amazing. I'm too dumb to code.  | Puja G           | 1-May-20    | 2           | python-data      | 4                |
| 14.        | Interesting overview.  | Dorottya B       | 30-May-18   | 4           | e-learning       | 4                |
| 15         | I'd love to learn such topics in future as well.   | Manisha o        | 29-Jul-20   | 5           | e-learning       | 5                |
| 16         | Excellent start for passionate educators ? those who want to make a difference in education systems. | By Nida A        | 1-Jun-20    | 4           | e-learning       | 5                |

match; both are equal to 4 stars.

In figure 13, we captured the user comment "great" in row 13 of table 6, which can be accessed through the link <https://www.coursera.org/learn/python-data/reviews?completers=true&sort=helpful&star=2>. This clearly shows the conflict between the user rating (2 stars) and our model's rating prediction (4 stars). The new predicted rating is more reasonable according to the sentiment analysis of the learner's comment.

## 8 Discussion

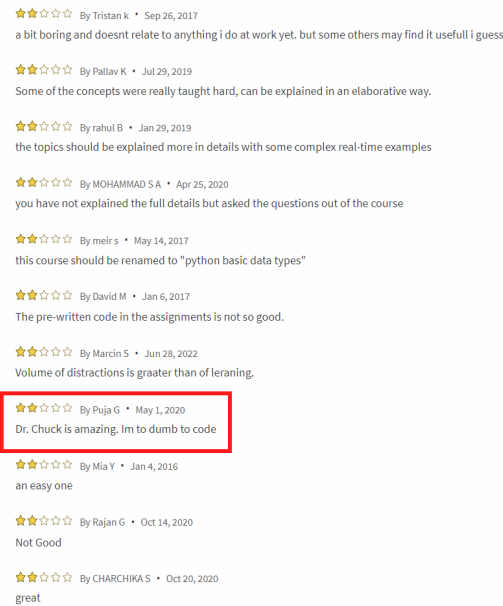
Table 6 displays a different sample of a new rating prediction made by a proposed model and shows the users' original ratings of the learning resource. Figures 12 and 13 are real-world samples from the Coursera site to support table 6. The results in table 6 and displayed user ratings and comments in figures 12 and 13 clearly explain that users' text reviews gave the recommender system greater efficiency and accuracy than rating. Additionally, we combined the users' ratings and reviews to get a filtered matrix factorization, directly influencing features extracting educational videos and learners' preferences, thus improving the recommender system. Table 6 contains four cases in which the proposed model presented a positive rating prediction over the original user rating:

- Case 1: The user wrote a positive review but gave a

low rating. A new predicted rating will have a high rate according to the learner's opinion in the text comment. For example, see rows 2, 6, 7 and 13 in table 6.

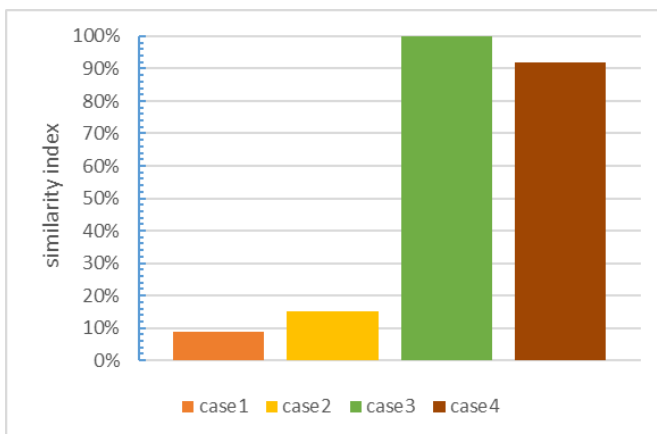
- Case 2: The user wrote a negative review but gave a high rating. The new predicted rating will be low according to the learner's opinion in the text comment. For example, see row 3 in table 6.
- Case 3: The user review and rating matched, regardless if positive or negative. A new predicted rating will be exactly equal to the original rating. For example, see rows 1, 5, 11, 12, 14 and 15 in table 6.
- Case 4: The user wrote a positive or negative review, and the rating was almost equivalent to the given opinion. The new predicted rating will be slightly different from the original rating but more expressive about the user's opinion because the sentiment analysis of the review takes into account all the information in the text review. For example, see rows 4, 8, 9, 10 and 16 in table 6.

As we can notice in Cases 1 and 2, a big conflict exists between the original and predicted ratings possibly caused by different reasons like a wrong rating if the user clicked on the wrong number. On the other hand, it could happen if the user was in a hurry to click the rating stars. From the presented results, we can show the degradation between the original users' ratings and the new predicted ratings, especially in cases 1, 2 and 4, so we measure the distance simi-



**Figure 13.** captured real sample of learner comment and feedback on python-data course

ilarity between the user rating matrix and the predicted rating matrix. That explains why using user data reviews will help to build a good learning RS more than using numerical rating data. Cosine similarity and TF/DF are used for similarity calculations Pandey [2018]. These calculations are made for all source datasets, which already contain different text reviews that vary in text length, size and type of content. In figure 14, we show a similar result for samples from each of the four cases.



**Figure 14.** similarity distance between original user rating and predicted rating in four case of new rating

In contrast, the similarity in the overall dataset was 98%, which means there is an incompatibility between learners' ratings and preferences reflected in the text reviews. This proves the difference can transform what is recommend or not recommend to the user. In other words, this changes the behavior of the recommender model.

## 9 Conclusion

We proposed a hybrid learning recommender system of rating and text-review data provided by learners in educational

videos. The proposed content-based learning recommender suggests the most suitable learning videos to learners per their interests. Our model runs in two parallel lines (input and output data lines). Each line comprises multiple phases: The input-line phases are learning data preparation, textual data preprocessing, and language model building, while the output-line phases are learners' historical data preparation, text-review sentiment analysis and new rating prediction, rating hybridization, and LFM. The last phase of the two lines will be used by the CNN model for the recommendation process. Using text reviews increased the accuracy of the recommendation system because this type of data provides more details about user interests, which can be employed to devise a good recommender system that meets user preferences. We used a hybrid of both types of historical user data (rating and comment) in the presented model to reduce cold-start problem effects. We used a real-world dataset to train and test the proposed model (70% training and 30% testing). This data came from the Coursera platform. Results showed that using online user reviews is more accurate than using numerical user ratings, which yields a high-performing recommendation system. For short term future works, we will use a transformer instead of a CNN. We tried to use a transformer but faced challenges due to its limitations with text sequencing, so we will look into how to deal with this.

## Declarations

### Authors' Contributions

Mounir Zrigui carried out the literature review. Manar Joundy developed the mathematical models. Mohsen Maraoui and Manar Joundy jointly conducted the simulations and drafted the manuscript. Moonir Ziguï provided useful remarks and critically reviewed the manuscript. All authors read and approved the final manuscript.

### Competing interests

The authors declare that they have no competing interests.

### Availability of data and materials

Data can be made available upon request.

## References

- Afoudi, Y., Lazaar, M., and Al Achhab, M. (2021). Hybrid recommendation system combined content-based filtering and collaborative prediction using artificial neural network. *Simulation Modelling Practice and Theory*, 113:102375. DOI: 10.1016/j.simpat.2021.102375.
- Ali, A. A. M. A., Mabrouk, M., and Zrigui, M. (2022). A review: Blockchain technology applications in the field of higher education. *Journal of Hunan University Natural Sciences*, 49(10). DOI: 10.55463/issn.1674-2974.49.10.10.

- Ayadi, R., Maraoui, M., and Zrigui, M. (2014). Latent topic model for indexing arabic documents. *International Journal of Information Retrieval Research (IJIRR)*, 4(2):57–72. DOI: 10.4018/ijirr.2014040104.
- Chen, L.-T., Liu, L., and Tretheway, P. (2022). Using multilayer videos for remote learning: Videos of session guidance, content instruction, and activity. *Computers in the Schools*, 38(4):322–353. DOI: 10.1080/07380569.2021.1989220.
- Dias, L. L., Barrère, E., and de Souza, J. F. (2021). The impact of semantic annotation techniques on content-based video lecture recommendation. *Journal of Information Science*, 47(6):740–752. DOI: 10.1177/0165551520931732.
- El Mabrouk, M., Gaou, S., and Rtili, M. K. (2017). Towards an intelligent hybrid recommendation system for e-learning platforms using data mining. *International Journal of Emerging Technologies in Learning (Online)*, 12(6):52. DOI: 10.3991/ijet.v12i06.6610.
- El Moustamid, A., En-Naimi, E., and El Bouhdidi, J. (2017). Integration of data mining techniques in e-learning systems: Clustering profil of lerners and recommender course system. In *Proceedings of the 2nd international Conference on Big Data, Cloud and Applications*, pages 1–4. DOI: 10.1145/3090354.3090453.
- Hazar, M. J., Maraoui, M., and Zrigui, M. (2022a). Recommendation system based on video processing in an e-learning platform. *Journal of Hunan University Natural Sciences*, 49(6). DOI: 10.55463/issn.1674-2974.49.6.6.
- Hazar, M. J., Toman, Z. H., and Toman, S. H. (2019). Automated scoring for essay questions in e-learning. In *Journal of Physics: Conference Series*, volume 1294, page 042014. IOP Publishing. DOI: 10.1088/1742-6596/1294/4/042014.
- Hazar, M. J., Zrigui, M., and Maraoui, M. (2022b). Learner comments-based recommendation system. *Procedia Computer Science*, 207:2000–2012. DOI: 10.1016/j.procs.2022.09.259.
- Herath, D. and Jayaratne, L. (2017). A personalized web content recommendation system for e-learners in e-learning environment. In *2017 National Information Technology Conference (NITC)*, pages 89–95. IEEE. DOI: 10.1109/NITC.2017.8285650.
- Hutto, C. and Gilbert, E. (2014). Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the international AAAI conference on web and social media*, volume 8, pages 216–225. DOI: 10.1609/icwsm.v8i1.14550.
- Islek, I. and Oguducu, S. G. (2022). A hierarchical recommendation system for e-commerce using online user reviews. *Electronic Commerce Research and Applications*, 52:101131. DOI: 10.1016/j.elerap.2022.101131.
- Kim, J., Park, J., Shin, M., Lee, J., and Moon, N. (2021). The method for generating recommended candidates through prediction of multi-criteria ratings using cnn-bilstm. *Journal of Information Processing Systems*, 17(4):707–720. DOI: 10.3745/JIPS.02.0159.
- Kulkarni, P. V., Rai, S., and Kale, R. (2020). Recommender system in elearning: a survey. In *Proceeding of International Conference on Computational Science and Applications: ICCSA 2019*, pages 119–126. Springer. DOI: 10.1007/978-981-15-0790-8\_13.
- Kundu, S. S., Sarkar, D., Jana, P., and Kole, D. K. (2021). Personalization in education using recommendation system: an overview. *Computational Intelligence in Digital Pedagogy*, pages 85–111. DOI: 10.1007/978-981-15-8744-3\_5.
- Legrand, A., Trystram, D., and Zrigui, S. (2019). Adapting batch scheduling to workload characteristics: What can we expect from online learning? pages 686–695. DOI: 10.1109/IPDPS.2019.00077.
- Mahmoud, A. and Zrigui, M. (2021). Semantic similarity analysis for corpus development and paraphrase detection in arabic. *Int. Arab J. Inf. Technol.*, 18(1):1–7. DOI: 10.34028/iajit/18/1/1.
- Murad, D. F., Heryadi, Y., Wijanarko, B. D., Isa, S. M., and Budiharto, W. (2018). Recommendation system for smart lms using machine learning: a literature review. In *2018 international conference on computing, engineering, and design (ICCED)*, pages 113–118. IEEE. DOI: 10.1109/ICCED.2018.00031.
- Pandey, P. (2018). Simplifying sentiment analysis using vader in python (on social media text). *Analytics Vidhya*. Available at: <https://medium.com/analytics-vidhya/simplifying-social-media-sentiment-analysis-using-vader-in-python-f9e6ec6fc52f>.
- Phongsatha, S. and Cleesuntorn, A. (2017). Elearning teachers' and students' perception toward live-video conference in an elearning environment. *AU eJournal of Interdisciplinary Research (ISSN: 2408-1906)*.
- Ricci, F., Rokach, L., and Shapira, B. (2021). Recommender systems: Techniques, applications, and challenges. *Recommender Systems Handbook*, pages 1–35. DOI: 10.1007/978-1-0716-2197-4\_1.
- Slimi, A., Hamroun, M., Zrigui, M., and Nicolas, H. (2020). Emotion recognition from speech using spectrograms and shallow neural networks. In *Proceedings of the 18th International Conference on Advances in Mobile Computing & Multimedia*, pages 35–39. DOI: 10.1145/3428690.3429153.
- Thakker, U., Patel, R., and Shah, M. (2021). A comprehensive analysis on movie recommendation system employing collaborative filtering. *Multimedia Tools and Applications*, 80(19):28647–28672. DOI: 10.1007/s11042-021-10965-2.
- Tolety, V. B. P. and Prasad, E. V. (2022). Hybrid content and collaborative filtering based recommendation system for e-learning platforms. *Bulletin of Electrical Engineering and Informatics*, 11(3):1543–1549. DOI: 10.11591/eei.v11i3.3861.
- Treude, C., Sicard, M., Klocke, M., and Robillard, M. (2015). Tasknav: Task-based navigation of software documentation. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, volume 2, pages 649–652. IEEE. DOI: 10.1109/ICSE.2015.214.
- Wei, C., Wang, Y.-C., Wang, B., and Kuo, C.-C. J. (2023). An overview on language models: Recent developments and outlook. *arXiv preprint arXiv:2303.05759*. DOI:

10.48550/arXiv.2303.05759.

- Wong, J. K., Oladinrin, O. T., Ho, C. M., Guilbert, E., and Kam, R. (2022). Assessment of video-based e-learning in a construction measurement course. *International Journal of Construction Management*, 22(1):1–7. DOI: 10.1080/15623599.2018.1435152.
- Xiao, J., Wang, M., Jiang, B., and Li, J. (2018). A personalized recommendation system with combinational algorithm for online learning. *Journal of ambient intelligence and humanized computing*, 9:667–677. DOI: 10.1007/s12652-017-0466-8.
- Yassine, A., Mohamed, L., and Al Achhab, M. (2021). Intelligent recommender system based on unsupervised machine learning and demographic attributes. *Simulation Modelling Practice and Theory*, 107:102198. DOI: 10.1016/j.simpat.2020.102198.
- Zhang, Y., Liu, R., and Li, A. (2015). A novel approach to recommender system based on aspect-level sentiment analysis. In *2015 4th National Conference on Electrical, Electronics and Computer Engineering*, pages 1453–1458. Atlantis Press. DOI: 10.2991/nceece-15.2016.259.