An-Najah National University

Faculty of Graduate Studies

IMPROVING ARABIC E-LEARNING USER EXPERIENCE THROUGH SENTIMENT ANALYSIS AND COLLABORATIVE FILTERING MODELS

By

Aya Said Yamin

Supervisor

Dr. Emad Natsheh

This Thesis is Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Artificial Intelligence, Faculty of Graduate Studies, An-Najah National University, Nablus, Palestine.

2025

IMPROVING ARABIC E-LEARNING USER EXPERIENCE THROUGH SENTIMENT ANALYSIS AND COLLABORATIVE FILTERING MODELS

By
Aya Said Yamin

This Thesis was Defended Successfully on 26/07/2025 and approved by

Dr. Emad Natsheh	
Supervisor	Signature
Prof. Mohammed Awad	
External Examiner	Signature
Dr. Amjad Hawash	
Internal Examiner	Signature

Dedication

This thesis work is dedicated to my beloved family, whose constant encouragement, support, and patience have been the base for this journey. To my husband, Eng. Mohammad Dwekat, for his continuous encouragement and understanding during the highs and lows in this journey. To my supervisor, for his guidance and faith in my abilities. And to everyone who inspired me to continue learning and pursuing knowledge.

Acknowledgements

All praise and thanks are due to Allah, the Lord of the worlds, and may the peace and blessings of Allah be upon the Seal of the Prophets and Messengers. And as Allah says, "And say, 'My Lord, increase me in knowledge." (Taha, 20:114).

Praise be to God, Lord of the Worlds, who provided me with the patience and strength to complete this academic thesis. Without His grace and blessings upon me, its completion would not have been an easy and simple matter.

I would like to thank my supervisor, Dr. Imad Natsheh, for walking with me on this journey and for his encouragement, guidance, and insightful comments, which have had a profound impact on the progress of the work.

I would like to express my deep gratitude to the faculty and staff of the Artificial Intelligence Department at An-Najah National University for providing a suitable academic learning environment.

To my family and friends, I would like to thank you for your prayers, continued support, and faith in me. You have been my source of inspiration and resilience.

From the bottom of my heart, I would thank everyone who had any connection to the completion of this dissertation.

Declaration

I, the undersigned, declare that I submitted the thesis entitled:

IMPROVING ARABIC E-LEARNING USER EXPERIENCE THROUGH SENTIMENT ANALYSIS AND COLLABORATIVE FILTERING MODELS

I declare that the work provided in this thesis, unless otherwise referenced, is the researcher's own work, and has not been submitted elsewhere for any other degree or qualification.

Student's Name: Aya Said Yamin

Signature: Aya

Date: 26/7/2025

List of Contents

List of Table

List of Figures

List of Appendices

IMPROVING ARABIC E-LEARNING USER EXPERIENCE THROUGH SENTIMENT ANALYSIS AND COLLABORATIVE FILTERING MODELS

By
Aya Said Yamin
Supervisor
Dr. Emad Natsheh

Abstract

This study works to develop an enhanced recommendation system (RS) for the purpose of improving the user experiences in e-learning environments by achieving the integration between sentiment analysis (SA) and collaborative filtering (CF). It uses a publicly available English dataset from Coursera and translates it into Arabic using AWS translation services to be appropriate for the target language context. It targets Arabic-language course reviews, taking into consideration handling data sparsity and linguistic complexity challenges. A refined multilingual Bidirectional Encoder Representations from Transformers (BERT) model was used to produce sentiment labels, and support vector machine (SVM) classification with a combination of Term Frequency–Inverse Document Frequency (TF-IDF) and FastText features achieved good performance. And then the user-item interaction matrix was enriched with the sentiment scores that resulted from SA to make recommendations using Alternating Least Squares (ALS) for active users and using K-means clustering based on profiles, followed by hybrid K-Nearest Neighbors (KNN) and TF-IDF similarity on Arabic course names for cold users. The evaluation of the system is done by making a comparison before and after adding the effect of sentiment separately for active and cold users. The system specifically targets users or learners who plan to study Arabic courses on online platforms. The findings emphasize that this integration of sentiment information reduces the limitations related to cold-start user problems and also enhances personalization. For example, the sentiment-aware model achieved a reduction in RMSE of nearly 70% for active users and a significant improvement in success rate (from 19.57% to 93.27%) and recall (from 18.68% to 91.24%) for cold users. These improvements lead to considering it a practical, useful solution for e-learning platforms.

Keywords: Sentiment Analysis, Collaborative Filtering, Recommendation System, E-Learning, Arabic Text, Cold-Start Problem, Multilingual BERT, SVM.

Chapter One

Introduction

1.1 Background and Context

Education systems evolved to meet the growing demands of societies as they developed [1]. As civilizations developed, establishments and educational institutions became hubs for the dissemination of knowledge [1]. The development of the printing press transformed education by making books more accessible, reducing illiteracy, and promoting the spread of knowledge [2]. This invention marked the beginning of the educational industrial revolution, which led to the creation of modern educational frameworks, the expansion of schools, and the founding of universities worldwide. In the end, this ongoing development cleared the path for the incorporation of science and technology into the classroom, revolutionizing the educational process [3].

E-learning has become essential to contemporary education over the past 20 years [4]. The easier access to the internet, developments in technology, and the increasing demand for flexible learning options have all helped the widespread use of digital education. A new coronavirus disease (COVID-19) appeared [5] in late 2019 and in 2020 spread on a worldwide scale, upending daily life in ways that were almost unthinkable, where educational institutions were forced to quickly adapt. Leading to the quickening of the shift to online education [6].

The rapid accessibility of this growing educational model sets it apart [5]. Geographical limitations and restrictions are lessened because learners can have access to e-learning resources from any location. Interactive components, such as audio and video, help reinforce cognitive knowledge and improve retention, making it more affordable.

However, it faces challenges [7], as shown in Table 1.1, especially in rural areas, due to unreliable internet connectivity and lack of access to digital equipment. Self-reliance, the ability to study independently, and the motivation and discipline to persist are also essential for this type of education. It is also necessary to take into account the caliber of the available e-content as well as the absence of suitable monitoring and assessment systems.

Table 1.1

E-Learning Obstacles

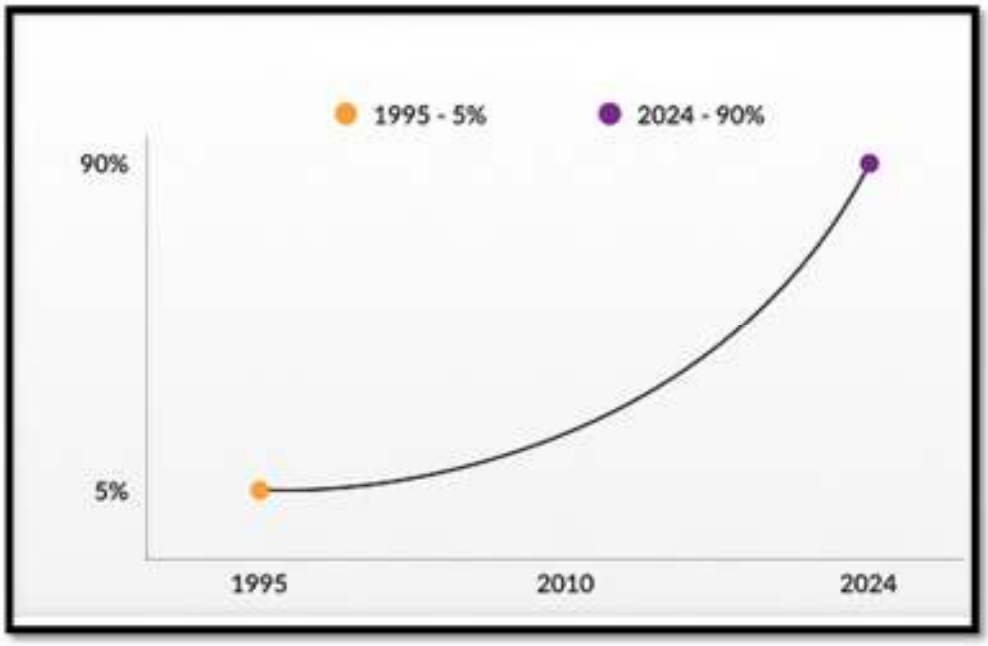
Category	Challenge	Description
Technical	Internet Connectivity	Education is hampered by the poor internet connection, especially in rural areas.
	Insufficient Access to Digital Devices	A significant number of students lack access to laptops, tablets, or cellphones.
	Usability of the Platform	E-learning platforms lack user-friendliness, complicating navigation.
Learner-Related	Self-Discipline and Motivation	Self-control and motivation can be difficult for students without direct interaction.
	Digital Proficiency	Students and instructors lack the digital skills required for effective e-learning engagement.
Curriculum and Instruction	Data Quality and Content Pertinence	Poorly organized or outdated content can affect educational outcomes.
	Restricted Interaction and Engagement	Lack of real-time interaction reduces student engagement.

As time passed, the demand for online education increased [3], as shown in Figure 1.1, leading to the rise of international platforms offering a variety of courses. These platforms cover a wide range of topics, from academic subjects to practical skills and even niche hobbies. There were online courses to suit students' demands, whether they wanted to learn a new language, become proficient in coding, comprehend difficult scientific ideas, or obtain qualifications in a particular job [1]. Over 70% of university students and a higher percentage of lifelong learners embraced this new form of education. However, the need for better user experiences on these platforms became clear.

Personalized course recommendations are key to enhancing the effectiveness of e-learning [9]. With so many online courses available, students often struggle to find material that aligns with their interests, skill level, and career goals [10]. Personalized RSs study and analyze the learners' past interactions, preferences, and learning trends using machine learning (ML) and artificial intelligence (AI) to recommend the best courses.

Figure 1.1

The Evolution of eLearning Adoption: An increase from 5% in 1995 to 90% in 2024, indicative of the swift expansion of digital learning, particularly propelled by technology innovations and worldwide transformations like the COVID-19 pandemic



Note: Adopted from [8].

These recommendations provide courses that match learners' interests, saving time that would otherwise be spent searching for relevant content [11]. This improves learning efficiency, boosts satisfaction and retention, and enhances skills and productivity.

SA and CF have been very helpful for personalized recommendations and personalized learning. The digital text analysis to ascertain whether the emotional tone is neutral, positive, or negative is opinion mining, or SA. But SA of Arabic text faces unique challenges related to Arabic's complicated morphology, rich structural signals, and dialectal variations [12]. Advanced natural language processing (NLP) methods are required to get rid of these challenges. These approaches comprise lexicon-based strategies, deep learning (DL) models, and hybrid methods that integrate grammar-based principles with ML [13].

CF is a recommendation method that produces recommendations depending on user interactions and activities [10]. This method depends on finding similarities between learners and content in order to specify trends and spot patterns in the behavior of users. In Arabic eLearning, there are some problems that CF may face, including limited user interactions, sparse data, and a lack of annotated Arabic datasets, because it could be

difficult to locate sufficient and well-structured user interaction data, which is important for CF efficacy.

However, combining CF and SA can lead to the improvement of recommendations and suggestions for Arabic courses [14]. SA can lead to user preference prediction improvements and generate more accurate and personalized course recommendations by depending on learners' comments, reviews, and engagement levels [15]. RSs may lead to enhancement of the overall e-learning experience by integrating these strategies in order to achieve a better and deeper understanding of learners' requirements, feelings, and preferences.

Therefore, this study focuses on addressing some of the challenges and building a sentiment-aware recommendation model especially designed for Arabic e-learning platforms that integrates SA with CF to improve the accuracy and personalization of course recommendations. The model will handle different types of users. It will capture deeper user preferences for active users by incorporating both sentiment scores and sentiment trends resulting from SA into the user-item interaction matrix. For cold-start users, a hybrid strategy that integrates clustering based on non-static profiles that are enriched with sentiments with content-based and collaborative filtering techniques. The efficacy of the model is evaluated using an Arabic-translated course review dataset, separately for both active and cold users, with a focus on better and deeper understanding of user preferences, improving early recommendation accuracy, coverage, and personalization.

1.2 Problem Statement

A major challenge facing Arabic recommendation systems is the lack of context awareness and bias [16]. These issues lead to unfair or inaccurate recommendations, often biased against specific subjects, dialects, or user demographics because many Arabic NLP models are trained on tiny datasets. In addition, Arabic sentences often, for context, depend on nuanced linguistics, which might be missing in the conventional SA techniques. For better improvements to SA and recommendation models, better preprocessing techniques, bigger, well-annotated datasets and sophisticated AI models that can deal with contextual subtleties and dialectal variances successfully are needed to take into consideration [17].

For better course recommendations and user satisfaction, there is a need for an RS that combines SA and CF [18]. While SA can extract insightful and useful information from users' reviews through the recordation of their thoughts and feelings about courses, CF recommends relevant content by employing user interactions. Combining these strategies allows the system to provide recommendations that are more accurate and tailored to the user's tastes, improving their educational experience. Sentiment-driven RSs outperform baseline CF recommenders by 30.7% in key metrics like Mean Average Precision (MAP), Normalized Discounted Cumulative Gain (NDCG), and precision@K [18].

The lack of accurate Arabic sentiment models is a major problem [16]. Even in more sophisticated models such as AraBERT, the accuracy across datasets varies from 84.65% to 92.13%, leading to the existence of noise in recommendation engines [19]. And because of these distinctions, a challenge is the integration of sentiment trends into Arabic e-learning RS.

According to these issues, it's obvious that in order to achieve a better RS, there is a need for sophisticated Arabic SA incorporation, trying to exceed the data constraints, and improving personalized learning experiences for a variety of users.

1.3 Research Objectives

This study aims to improve user experience in Arabic courses on e-learning platforms by integrating SA with CF. Among the primary objectives:

1. Develop and execute an ML-driven Arabic SA model to assess and categorize course reviews with high performance.
2. Develop and compare two CF RSs for e-learning course recommendations:

- Develop a sentiment-aware RS by adapting the ALS algorithm to incorporate sentiment scores into the user-item interaction matrix for active users, modifying course ratings according to sentiment polarity and trends. And a CB, or clustering-based method, for cold start users depending on sentiment-enhanced user profiling in order to produce recommendations.
- Develop a baseline conventional CF model utilizing standard ALS for active users and the traditional CB, or clustering-based method, for cold start users, without incorporating sentiment integration, to serve as a benchmark.

3. Study the effect of sentiment integration on course relevance and user engagement by employing a detailed comparison between baseline recommendation and sentiment-aware recommendation models for both active and cold users using performance metrics, such as Root Mean Square Error (RMSE), Precision@K, Recall@K, NDCG, Diversity, and Coverage.

1.4 Research Questions

1. How does SA improve the accuracy of course recommendations for Arabic e-learning courses?
2. What is the comparison between sentiment-aware CF and traditional models in forecasting user preferences for Arabic courses?
3. How do user engagement, satisfaction, and recommendation efficacy affect when SA is added to CF for Arabic e-learning content?
4. What is the significance of sentiment-based user profiling in enhancing cold-start recommendations for Arabic courses?

1.5 Contribution

The Coursera Course Reviews dataset was collected [20]. It is divided into two subsets:

- Coursera courses: This subset lists all 622 of the courses that are available on Coursera.
- Coursera Reviews: This subset includes 1.45 million reviews and ratings.

From this original dataset, 20,000 reviews were taken. At first, a stratified sampling approach was used to build a base sample by picking up to 10 reviews per course to focus on the diversity across different courses. Then, from the remaining data, additional reviews were randomly sampled, making sure there was no duplication of previously selected reviews. This process guarantees variety in course representation and reviewer feedback.

For the final sampled dataset of 20,000 English reviews, Amazon Translate from Amazon

Web Services (AWS) was employed for translation into Arabic to support the aim of the study.

The process of achieving a high-quality Arabic dataset from the original English dataset was done through a structured pipeline. At first, the English CSV file from an AWS S3 bucket was extracted and loaded into a Pandas DataFrame for processing. Then, an automated translation mechanism was developed using the translate_text() function, which for each review sent API requests to AWS Translate. Batch processing was applied to handle the large dataset effectively to boost efficiency. Then, the translated Arabic text was reintegrated into the original DataFrame structure with the same intact. Finally, the translated dataset was stored back into S3 for further analysis and processing in the later stages.

1.6 Thesis Structure

The chapters that made up this thesis were divided into multiple divisions. There were four chapters and a conclusion following the introduction chapter. Previous studies on SA, CF, and RSs in e-learning are covered in Chapter One. The dataset, preprocessing methods, and model construction procedure are covered in Chapter Two. The results and analysis of the evaluation are presented in Chapter Three. Key findings are outlined in Chapter Four along with recommendations for future study directions.

1.7 Literature Review

With the evolution of the digital world, there is an increasing need to improve the user experiences in various sectors, not only learning. Therefore, this section offers a selection of relevant research studies that examine different facets of user experiences in diverse circumstances. This section aims to give a summary of these relevant studies that cover the main pillars of this study: SA, RS, and hybrid approaches (SA + RS).

1.7.1 Sentiment Analysis

The study [21] investigates how students see education after COVID-19 by developing Support Vector Machine Sentiment Analysis for Arabic Students' Course Reviews, SVM-SAA-SCR, an SA system that categorizes Arabic course reviews from Prince Sattam bin Abdulaziz University (PSAU) using C-Support Vector Classification (SVC). This system tries to address issues such as shortage of resources, dialectal variations, and morphological complexity. The Arabic text before the classification went through a stage of preprocessing steps, including normalization, stemming, tokenization and stopword removal. Then, a TF-IDF vectorization was used, although it exhibits a lack of

semantic comprehension. With 84.7% overall accuracy and 69.62% positive sentiment accuracy, the system closely matches CAMeLBERT's 70.48% in positive sentiment. Nonetheless, SVM encounters difficulties with contextual issues, including negation. This method can help raise the standard of education while offering insightful information on the difficulties Arab colleges face.

Some studies develop a multi-criteria approach for Arabic dialect SA, as study [22] intends to concentrate on internet evaluations. With an emphasis on Saudi dialect reviews, the authors present a multi-criteria assessment system for evaluating and ranking ML classifiers for Arabic SA. With a variety of performance metrics, including precision, recall, and CPU time, five common ML classifiers were used and evaluated, such as DL, SVM, Decision Tree (DT), Naive Bayes (NB), and KNN. Tokenization, stopword elimination, and stemming are examples from the preprocessing stage. SVM and DL classifiers achieved the best balance across metrics, with the highest accuracies of 85.25% and 82.30%, emphasizing the importance of multi-criteria evaluation for SA in morphologically complex languages like Arabic.

The complex morphology of Arabic, dialectal variations, and scarcity of sentiment resources make SA incredibly challenging. Using supervised learning (SL), unsupervised learning (UL), and hybrid ML techniques, the study [23] investigates the SA of Arabic tweets. The authors labeled over 2,000 Arabic tweets manually using preprocessing steps including tokenization, normalization, stopword removal, and stemming. And they use the N-grams, TF-IDF, and lexical sentiment terms as extracted features in order to compare SL models, the SVM with NB. Moreover, they built a hybrid model and a UL lexicon-based classifier. SVM with TF-IDF achieved the highest accuracy of 78.08%, outperforming NB with 74.8%. Expanding the sentiment dictionary improved the lexicon-based method to 75%, while the hybrid model failed to surpass a 73.45% F-measure and 64.33% accuracy. The study concludes that SVM is most effective for Arabic SA, while lexicon-based methods need ongoing refinement for dialectal variation.

The research in [24] uses SL ML to categorize Arabic newswire comments from Echorouk Online into positive, negative, and neutral feelings. Multiple preprocessing techniques are employed, such as uniform resource locator (URL) elimination, special character and punctuation filtering, tokenization, normalization, stopword removal

(excluding negation terms), and light stemming (removing prefixes and suffixes without modifying word meaning). The study employs unigrams and bigrams with Bag of Words (BOW) and TF-IDF representations to evaluate six ML classifiers (Multinomial Naive Bayes (MNB), Linear SVC (LSVC), Random Forest (RF), Logistic Regression (LR), Multi-layer Perceptron (MLP), and KNN); the results show that MNB achieves the highest accuracy (85.57% binary, 65.64% three-class); the n-gram integration increases accuracy by ~10%, emphasizing that n-gram features significantly improve Arabic SA, count vectors outperform TF-IDF, and binary classification is more effective.

In the field of Arabic SA, the paper [25] highlights the difficulties related to language complexity, dialects, and limited resources and suggests an ensemble DL model combining Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) to classify tweets into positive, negative, and neutral sentiments. Then, they do preprocessing by removing noise, standardizing language, and applying the AraVec word embeddings. The ensemble model achieves a 64.46% F1-score on the Arabic Sentiment Tweets Dataset (ASTD), outperforming the Recursive Neural Tensor Network (RNTN) model's 53.6%. Results show that either CNN or LSTM performs well individually, but their combination improves accuracy. The study comes to the conclusion that although DL greatly improves Arabic SA, more sophisticated architectures, remote training, and bigger datasets could still lead to improvements.

In order to ascertain the emotional polarity of student feedback, the study [26] investigates the use of SA in e-learning. Finding the best n-gram model (unigram, bigram, or trigram) for feature extraction in conjunction with ML methods like SVM and Hidden Markov Model (HMM) is the goal of the study. The study applies feature selection methods to improve the accuracy of classification, such as Mutual Information (MI), Chi-Square (CHI), and Information Gain (IG), on e-learning data from Udemy, NPTEL (National Programme on Technology Enhanced Learning), and SWAYAM (Study Webs of Active Learning for Young Aspiring Minds) platforms. The use of bigram and trigram models significantly increases performance; the highest F1-score of 0.803 was achieved with IG-based selection. The results ensure that SA leads to enhanced user experience by providing insights about student perspectives.

The study [27] examines using SA to improve e-learning platforms by analyzing 2,000 user comments from blogs and forums. The authors apply feature selection methods (IG, MI, CHI) to increase sentiment classification accuracy. They propose a hybrid SVM-HMM model. After using n-grams and preprocessing with stemming and stopword removal, the model with the sum rule achieves the best results, with IG performing better than MI and CHI. The study highlights challenges like informal language, misspellings, and noisy text and ensures that SA can provide insightful information about user feedback.

The paper [28] examines the sentiment classification of Arabic tweets that pertain to the learning experiences of university students in order to examine the correlation between student sentiments and their educational experiences, employing NB and SVM. Also, to address the morphological complexity and colloquial variations of Arabic, the study collected 2,000 Arabic tweets from King Abdulaziz University's e-learning platform and employed a variety of Arabic text preprocessing techniques, such as stop word removal, stemming, tokenization, normalization, and the handling of emoticons, punctuation, and elongated words. The feature extraction process was conducted using TF-IDF with n-gram features, which demonstrated that SVM outperformed NB by achieving 84.62% accuracy in binary classification (positive vs. negative) and 73.15% accuracy in three-class classification (positive, negative, neutral). The robustness of classification was enhanced by the incorporation of neutral sentiment, as evidenced by the Receiver Operating Characteristic (ROC) curve analysis, which confirmed the superiority of SVM. The paper emphasizes the difficulties associated with Arabic SA, including the absence of comprehensive sentiment lexicons, dialectal variations, and a complex morphology.

The research [29] investigates the impact of Arabic SA on online learning during COVID-19. It employs NLP and ML to extract emotions and classify sentiment polarity.

The National Research Council lexicon is employed to analyze emotions after the Arabic messages are preprocessed using Twitter APIs, which include the handling of Franco-Arabic words, subjectivity filtering, orthographic normalization, and negation detection. The study applies different ML classifiers, including NB, LR, KNN, and SVM, with SVM achieving 89.6% as the highest accuracy. IG-based feature selection

enhances classification by removing superfluous words. The study specifies that in online learning, the main causes of negative sentiment are technical issues, absence of supervision, and digital expertise, while emotion analysis shows feelings of anger, diversity, and limited resources and shows how the approach can help in improving experiences of e-learning and management of crises.

1.7.2 Recommendation System

The study [30] suggests a collaborative course RS using sequential pattern mining with the Sequential Pattern Discovery using Equivalence Classes (SPADE) and Association Rule Mining (ARM) with the Apriori on HarvardX and MITx data to study the relationships between students and courses. K-means clustering was used to improve recommendation quality by clustering students based on course history and student academic achievement. The stage of data preparation includes missing data handling and course sequence management. The process of evaluation was done by making a comparison with and without clustering. The results show that while SPADE better represents learning sequences (more accurate) than using only the Apriori, clustering enhances efficacy, generalization, and rule coverage, leading to more personalized and successful course recommendations.

A course RS was suggested in study [31] in order to address issues including inadequate decision-making and course overload by matching academic goals with students' interests. The system applies memory-based CF with the Pearson correlation coefficient (PCC) using Kaggle data in order to find top-K similar students based on course performance. The preprocessing includes handling missing values and one-hot encoding. Top-T option courses are recommended based on the weighted average of neighbors' scores, which is used for course score prediction. Furthermore, clustering and ARM improve recommendations, leading to the reduction in dropout rates and academic risks, while CF increases relevance and accuracy.

A cross-domain personalized RS was suggested in study [32] for the goal of improving the online educational materials in English using LR, CF, and ML. The system, in order to improve recommendation precision, employs matrix factorization, domain-specific features, and cross-domain information. It makes use of learning history, demographic data, and behavioral variables in order to assign probability scores to content items and

improve suggestions through learning and user feedback. A comparison was made between the baseline and three scenarios, showing the improvements in diversity, novelty, accuracy, and user satisfaction. Regression analysis reveals that education and age, in a big way, affect recommendation relevance. Customized lists of content are generated based on each learner's preferences and skill level. Simulation results show that the system enhances learning outcomes and engagement, providing a flexible, scalable, and AI-driven solution for cross-domain content recommendation and adaptive e-learning.

The study [33] proposes a data mining-based personalized web-based e-learning RS to improve user learning. It analyzes user behavior and content needs using K-means clustering and evolutionary algorithms to improve search results. The system recommends based on browsing history, contextual needs, and content relevancy for personalized learning. The Tri-Factor Recommendation (TFR) system uses web access logs, web usage mining, and user behavior tracking to analyze behavior using KNN and evolutionary algorithms to optimize suggestions. Log file analysis, feature extraction, keyword-based relationship modeling, and a hybrid search approach using KNN-refined search results and a genetic algorithm (GA) improved suggestion quality. The design of this system includes an administrator, an e-learning server, and learners to back up recommendations for adaptive resources. Based on performance evaluation, TFR performs better than standard RSs in terms of error rates of 10.6–22%, memory usage (22,618–24,674 kilobytes), processing time of 269–512 ms, and accuracy of 78–89.4%. The results show that the system enhances recommendation efficiency, precision, and scalability, making it ideally suited for personalized, video-based e-learning environments.

A hybrid DL model with CNN, Residual Network (ResNet), and LSTM was used in study [34] to improve online course RSs, addressing the challenges related to course selection in e-learning platforms, where the suggestions made are based on academic history, student behavior, and learning preferences. Preprocessed student data, DL architectures extract key patterns, and CNN, ResNet, and LSTM probability fusion combine strengths. The model was tested on 750 student records across courses of varying complexity (low, medium, and high), resulting in the model performing better than CNN with 93.2%, LSTM with 96.87%, and ResNet with 96.87%, achieving

accuracy of 99.2% with the Adam optimizer. This approach makes enhancement of user satisfaction, decision-making, and course recommendations, whereas dropout rates are reduced by adaptive learning path recommendations based on engagement patterns.

The study [35] introduces an RS designed for the E-Dirassa platform to make improvement for course recommendations, addressing challenges such as over-specialization and the cold-start problem. This approach integrates three methodologies, including CB filtering, course-based CF (CBCF) for active learners, and static profile-based CF (SPBCF) for new learners. CB suggests courses by employing textual similarity metrics, with the Dice coefficient recognized as the optimal measure through multi-criteria decision assistance (MDA) methodologies. CBCF discerns commonalities among courses by analyzing learners' historical interactions, utilizing a binary evaluation matrix and cosine similarity for recommendations. Euclidean distance is used to analyze analogous static profiles (e.g., age, academic level, language, and specialty) to make beginner recommendations. Integrating methods improves recommendation precision and breadth while resolving over-specialization and cold-start issues, according to the research. The authors note that the system's non-intrusive features eliminate the need for direct user input for recommendations, improving efficiency and usability.

The study [36] examines an innovative approach to web-based educational RSs, addressing the cold-start problem using coupled learning and bootstrapping. While conventional CF challenges new users and items due to a lack of prior interaction data, CB offers instant recommendations but runs the risk of overspecialization. To mitigate these restrictions, the authors offer an incremental semi-supervised learning (SSL) methodology, wherein various ML algorithms incrementally enhance their training by swapping examples categorized with high confidence, a technique derived from co-training. This technique enables the RS to incrementally augment its labeled dataset, hence improving recommendation quality for customers with less contact history. The methodology was assessed using Moodle-based e-learning data, revealing that the coupled-learning model attained performance levels similar to traditional methods while necessitating far fewer labeled cases initially. These findings underscore the efficacy of SSL in enhancing customization in educational RSs, especially in contexts with little

user interaction data, hence providing a scalable and adaptive resolution to the cold-start issue.

The work [37] introduces a hybrid approach that integrates knowledge-based (attribute-based) filtering with CF to mitigate the cold-start issue in e-learning. This approach, in contrast to conventional CF that depends on previous user ratings, formulates learning recommendations by utilizing user and content traits, integrating learning styles, preferences, and prior knowledge to tailor suggestions. The Rogers-Tanimoto metric was used in the model for content and user comparison, and the Jaccard similarity for user similarity. A web-based prototype evaluation showed the system's ability, even in complete cold-start scenarios, to identify similar users. The findings indicated an 82% satisfaction rate for suggested resources, 16% higher than existing cold-start systems, and a 90% satisfaction rate for student profile accuracy. With the use of ARM with the Apriori, the model can modify material qualities according to user evaluations. The study suggests that the hybrid RS enhances learner modeling, content suggestions, and user similarity recognition, providing a viable remedy in online education for addressing cold-start issues.

1.7.3 Hybrid Recommendation System

A hybrid RS was presented in study [38], integrating CF into SA from Twitter, leading to the improvement of course recommendations. SentiWordNet, TextBlob, VADER (Valence Aware Dictionary and sEntiment Reasoner), and TF-IDF were used to classify sentiment using SVM (the most effective), NB, and RF. The sentiment scores were included in KNN using PCC, allowing the system to combine explicit ratings with user sentiment. Results show that incorporating SA relieves data sparsity and the gray-sheep problem, improving precision, with RMSE decreasing from 0.689 to 0.683.

The paper [39] presents a sentiment-enhanced RS for Arabic literature, enhancing suggestion accuracy by combining CF with SA. The study preprocesses and categorizes sentiment with the Large-scale Arabic Book Reviews (LABR) dataset and employs Mazajak, Arabic BERT-Mini, and AraBERT (Arabic Bidirectional Encoder Representations from Transformers), with Arabic BERT-Mini attaining the maximum accuracy in sentiment classification. SA is incorporated by adjusting user ratings according to sentiment scores, so ensuring that recommendations account for both

explicit numerical ratings and implicit opinions derived from textual evaluations. The modified ratings are subsequently integrated into memory-based (user-user, item-item) and model-based (Singular Value Decomposition (SVD), kNN, and Non-negative Matrix Factorization (NMF)) CF techniques to improve recommendation efficacy. The performance assessment utilizing RMSE and Mean Absolute Error (MAE) indicates that SVD combined with Arabic BERT-Mini SA yields the optimal results (RMSE = 0.580, MAE = 0.42), markedly enhancing accuracy. The results underscore the necessity of using SA in Arabic RSs to alleviate cold-start issues and improve suggestion efficacy.

The study [40] presents a comprehensive methodology for improving RSs through the incorporation of SA using hybrid DL models alongside CF. The article identifies the constraints of conventional CF approaches, including sparsity and cold-start issues, and presents an adaptive framework that integrates user sentiment derived from reviews to enhance suggestion precision. The methodology utilizes CNN-LSTM and LSTM-CNN architectures, augmented by BERT embeddings for sentiment categorization, proficiently converting textual assessments into significant feedback. The integration approach amalgamates sentiment-driven predictions with CF models (SVD, NMF, SVD++) via a weighted combination utilizing a configurable β parameter. With β = 0.3 maximizing rating accuracy (lower RMSE, Normalized Mean Absolute Error (NMAE), and MAE) and β = 0.7 improving top-N recommendations (higher MAP, Mean Reciprocal Rank (MRR), and NDCG), this parameter leads to the balance between sentiment-based predictions and traditional CF ratings. The evaluation was done using two datasets, showing that sentiment-aware models outperform traditional CF, improving data sparsity and implicit preference capture, also showing a significant reduction in prediction errors and an improvement in recommendation dependability.

The research [41] introduces a college course RS, that incorporates SA and ML to improve

course selection. Student input is gathered via surveys, employing feature extraction methods such as TF-IDF and N-gram to discern pertinent keywords. The integration procedure commences with the extraction of course-related keywords from student replies, succeeded by SA to identify the most favorably regarded themes. Themes are aligned with available courses using KNN for fuzzy logic for rule-based classification

and similarity-based suggestions. The evaluation of the system was done with real student feedback, and it was presented that combining fuzzy logic with N-grams and TF-IDF achieved an accuracy of 85.71% in predicting appropriate courses. The study ensures qualitative data analysis value and suggests some future enhancements, such as latent Dirichlet allocation-based (LDA) topic modeling and a more interactive real-time recommendation interface.

The study [42] offers a multilingual RS that incorporates SA to aid Algerian users in selecting products, restaurants, and movies based on internet reviews. The objective is to integrate RSs with SA to enhance precision. The system first performs SA using semi-supervised SVM (S3VM) to classify customer reviews as positive, negative, or neutral by identifying key linguistic features such as word polarity, adjectives, adverbs, predicates, emotionality, and reflexivity. Polarity scores are assigned to the classified reviews, which are then included in a user-based CF algorithm using Spearman similarity for the recommendation of items based on sentiment-enhanced preferences. This approach leads to recommendation accuracy improvements, even in the absence of explicit ratings, by using both unlabeled and labeled evaluations, unlike traditional CF approaches. The evaluation of many datasets (English, French, Arabic) exhibited high precision (96-100%) and recall (100%), confirming that sentiment-based filtering significantly enhances recommendation quality. This study demonstrates that combining SA with CF alleviates the limitations of both approaches, leading to more customized and relevant recommendations.

The study [43] examines a hybrid RS for the e-learning domain that combines SA with CF to improve recommendation precision. It uses unsupervised DL to cluster learners by learning style and preferences, then applies an SA model that does processing for user comments by text cleaning, normalization, tokenization, and TF-IDF weighting to calculate resource sentiment. The integration is achieved by modifying the learner-item correspondence matrix, eliminating adversely received content, and emphasizing items with favorable feedback. The technology analyzes extensive datasets from Massive Open Online Courses (MOOCs), integrating student interaction data with course discussions to improve suggestions. Findings demonstrate that the integration of sentiment-based filtering enhances the pertinence and efficacy of recommended educational resources.

The study [44] introduces an enhanced RS in e-learning that integrates SA with rating-based methods, addressing data sparsity and cold-start issues. It makes use of feature extraction techniques like TF-IDF, LDA, Latent Factor Model (LFM), and word embeddings on video content and learner reviews before using SA. Sentiment Analyzer (Vader) was used for assigning sentiment-based ratings (1–5), which may enhance predictions by being integrated into the initial rating matrix. By combining explicit and sentiment-inferred ratings, a CNN with matrix factorization enhances user-item interactions. The model was tested on Coursera reviews, achieving a 0.98 cosine similarity, showing strong alignment between recommendations and user preferences.

This study [45] examines the incorporation of SA into RSs in e-learning platforms to improve user experience. The authors present a new recommendation model, SABCNN (Sentiment Analysis-Based Convolutional Neural Network), which employs NLP and DL methods to assess learners' sentiments derived from reviews and ratings of e-learning content. This study includes gathering and preprocessing user reviews and specifying review polarity by employing word embeddings and using CNN-based sentiment classification. Personalized learning resource recommendations depend on predicted sentiment scores. Five sentiment levels were used to categorize a dataset of Amazon book reviews for the evaluation of various CNN models. To highlight the multi-level sentiment classification importance in personalized recommendations, the CNN two-channel model, which combined pre-trained Global Vectors for Word Representation (GloVe) embeddings with fine-tuning, gained 0.77 as the greatest accuracy and decreased overfitting.

The Enhanced e-Learning Hybrid RS presented in [46] enhances e-learning recommendations through the integration of adaptive profiling and SA. It addresses cold-start, data sparsity, and customization challenges by dynamically creating learner profiles based on browser behavior and semantic connections utilizing DBpedia and WordNet ontologies. For better representation, Skip-Gram (S-G) and Continuous Bag of Words (CBOW) embeddings with a CNN-based SA model evaluate user reviews and forecast ratings. The recommendation engine to improve recommendations uses updated student profiles and predicted ratings. The CNN performs better than traditional ML techniques with an accuracy of 89.1%. The study concludes that by integrating learner

preferences, semantic profiles, and SA-driven ratings, the system dramatically improves content relevance and accuracy.

The study [47] suggests RS for e-learning platforms integrating user reviews and ratings to improve course selection. It combines CNNs for SA, LDA for topic modeling, and LFMs for feature extraction. Sentiment-predicted ratings from reviews are combined with explicit ratings and standardized to create a single rating matrix to improve recommendation quality. In the case of the absence of explicit ratings, the system predicts similar courses or users using sentiment scores and CF and CB. It's evaluated using Coursera data, combining structured and unstructured feedback, boosting recommendation precision, and personalization.

Table 1.2

Summary of research pertinent to SA

Ref	Dataset	Lang.	Feature Extraction	Sentiment Analysis	Evaluation
[21]	Student Reviews from PSAU's	Arabic	TF-IDF	SVM	Accuracy=84.7%
[22]	Twitter	Arabic	-	DL, SVM, DT, NB, KNN	Accuracy = 85.25% (DL)
[23]	Twitter	Arabic	N-grams, TF-IDF, Sentiment Lexicons	SVM, NB, Lexicon-based, Hybrid	Accuracy = 78.08% (SVM, TF-IDF)
[24]	Comments scraped from Echorouk Online	Arabic	N-grams, TF-IDF, BoW	MNB, SVM, RF, LR, MLP, KNN, LSVC	Accuracy =85.57% (MNB, binary classification)
[25]	ASTD	Arabic	AraVec	ensemble model (CNN, LSTM)	Accuracy = 65.05% F1-score 64.46%
[26]	Reviews (Udemy, NPTEL and Swayam)	English	N-grams, IG, CHI, MI	HMM, SVM	F1-score = 80.3% (SVM)
[27]	e-learning reviews collected from blogs and forums	English	N-grams, IG, CHI, MI, TF-IDF	hybrid (HMM, SVM)	F1-score = 80.3% (IG)
[28]	Twitter	Arabic	N-grams, TF-IDF	NB, SVM	Accuracy = 84.62% (SVM)
[29]	Twitter	Arabic	IG	NB, LR, SVM, KNN	Accuracy = 89.6% (SVM)

Table 1.3

Summary of research pertinent to the RS

Ref	Dataset	Lang.	Recommendation System	Evaluation
[30]	HarvardX and MITx	English	Hybrid (K-Means clustering, Apriori, SPADE)	Coverage = 59.4% (SPADE, Clustering)
[31]	Kaggle	English	Memory-Based CF (PCC, KNN, Score Prediction)	Hit Rate ≈ 70%, Accuracy = 0.8 to 1, Precision = 0.95, Recall = 0.5
[32]	Online teaching content	English	CF (Item-Based & User-Based (PCC, Cosine), CB (LR), Hybrid	Accuracy = 85%, Diversity = 0.72, Novelty = 0.78, User Satisfaction = 0.82
[33]	E-learning dataset	English	Hybrid (K-Means Clustering, KNN, GA)	Accuracy = 89.4%, Error Rate = 10.6%
[34]	Student-course interactions from Moodle	English	HDL (CNN, ResNet, LSTM)	Accuracy = 99.2%, Precision = 99.2%, Recall = 99.2%, F1-score = 99.2%
[35]	E-Dirassa e-learning platform	English	CB (Dice Coefficient), CBCF (cosine similarity), SPBCF (Euclidean Distance)	WSM score = 1.8889 (Dice Coefficient)
[36]	e-learning data from Moodle	English	Hybrid (CF, CB, SSL)	Accuracy = 73% (NB)
[37]	65 learning materials	English	Hybrid (knowledge-based filtering, CF (Jaccard similarity))	MAE = 1.05, Satisfaction Rate = 82%, Accuracy = 90%

Our work presents an innovative Arabic e-learning RS that combines SA with CF to enhance course recommendations and adeptly address the cold-start issue. In contrast to earlier studies that depend exclusively on explicit user ratings or CF, our methodology integrates Arabic sentiment-based features derived from course reviews, tackling the complexities of Arabic text processing, including intricate morphology, diacritics, and negation management. We utilize a preprocessing pipeline incorporating Camel Tools' MLEDisambiguator and Morphological Tokenizer for morphology-sensitive tokenization, diacritic eradication, stop word removal, normalization, and negation management to improve text quality. We employ a mix of TF-IDF and FastText

embeddings with an SVM classifier for sentiment classification, attaining excellent accuracy in sentiment polarity identification, while sentiment trends are analyzed by linear regression to monitor shifts in user preferences over time. We improve ALS-based CF for active users by applying a weighted interaction score that combines sentiment scores from user reviews and sentiment trend analysis with traditional recommendation signals, such as ratings, to produce personalized course recommendations. We introduce a hybrid K-means and kNN methodology for cold-start users, where CF approaches typically fail because of limited historical interactions. Initially, we implement K-means clustering to categorize users according to sentiment scores, engagement behaviors, and course preferences. Subsequently, we employ kNN inside the cluster to identify the most analogous users and suggest courses based on both user-based and item-based kNN similarity. Additionally, we employ TF-IDF similarity on Arabic course names to suggest semantically relevant courses that exhibit elevated sentiment ratings and favorable feedback trends. We do a comprehensive evaluation of the proposed approach with customized assessment metrics for users who are active and inactive. Precision@K, Recall@K, RMSE, and NDCG@K are used to evaluate performance for active users to ensure precise and optimally ranked course recommendations. Precision@K, Recall@K, Success Rate, Coverage, and Diversity are employed for cold users, ensuring that recommendations stay relevant, tailored, and diverse even for users with few interactions. Our sentiment-enhanced RS is adaptable, scalable, and perfectly suited for Arabic e-learning systems due to its dual evaluation process, which supplies an understandable and balanced evaluation of the system.

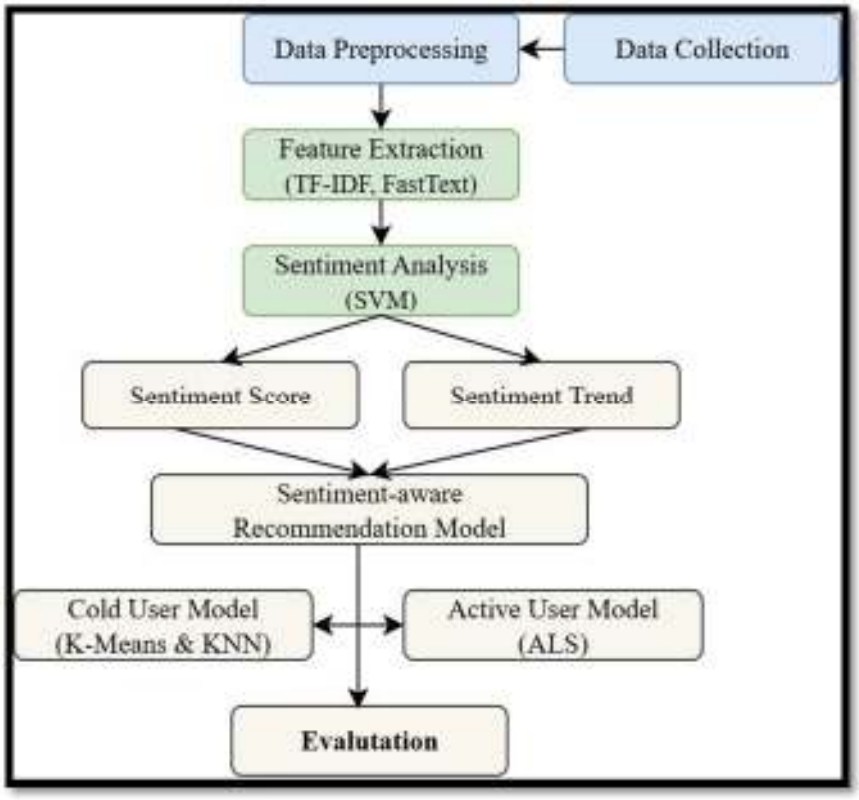
Chapter Two

Methodology

This chapter outlines the methodology adopted in the development of the proposed system. As illustrated in Figure 2.1, the system is structured into five main stages: data collection, data pre-processing, feature extraction, sentiment analysis, and the recommendation module. Each stage plays a critical role in transforming raw user data into meaningful recommendations. In the sections that follow, each component will be discussed in detail, highlighting the techniques, tools, and algorithms used at every step.

Figure 2.1

System Architecture of the Sentiment-Enhanced Hybrid RS



The methodology of work begins with the collection of e-learning data and preprocessing it to help in normalizing and cleaning reviews of Arabic courses. Then, hybrid feature extraction using TF-IDF and FastText is used to represent textual data. In the sentiment analysis, the SVM classifies the Arabic reviews into sentiment classes and produces sentiment scores as output to be used in the recommendation model. These sentiment scores, with some sentiment features, such as sentiment trend, are integrated to create a sentiment-aware recommendation model within a collaborative filtering

framework. Active and cold users are treated separately; for active users, the ALS is used, whereas for cold ones, K-means clustering and KNN-based are used. Finally, the performance of the model is evaluated using various metrics for both active and cold users separately.

2.1 Data Collection and Pre-processing

The dataset utilized in this study was obtained straight in CSV format. No supplementary crawling or external data acquisition techniques were necessary. The dataset comprises two files: Coursera courses and Coursera reviews.

This dataset was chosen for its relevance and pertinence to the research objective, which aims to improve e-learning course recommendations through SA, which made a sturdy basis for both the sentiment classification and RS of the study. Therefore, a subset of the dataset was translated into Arabic using Amazon Translate to aid the aim of the study.

The way of sentiment labeling for the Arabic course reviews was done automatically using a transformer-based approach to enable sentiment-aware recommendations. This approach ensures the label quality and reliability by the combination of pre-trained models, fine-tuning, and rating-based validation.

A. Preliminary Sentiment Classification Utilizing BERT

This approach of labeling initiated with the application of a pre-trained multilingual BERT model, namely, nlptown/bert-base-multilingual-uncased-sentiment, which facilitates numerous languages, including Arabic. This model, accessible via the Hugging Face Transformers library, can assess sentiment on a star scale (1-5).

This model, via the Hugging Face SA pipeline [48], was used to process each Arabic review. Then, the star ratings that resulted were associated with sentiment categories as outlined below:

- Positive: (4, 5) stars
- Neutral: 3 stars
- Negative: (1, 2) stars

The classification was recorded in a new column titled sentiment_label, denoting the sentiment created by BERT for each review.

B. Refinement for Domain Adaptation

The pre-trained model showed satisfactory performance [49], even though it was not trained particularly on Arabic e-learning content. The model was further refined on the available dataset of translated Arabic reviews to enhance domain adaptation using the original star ratings as weak labels. The AraBERT model (aubmindlab/bert-base-arabertv02) was chosen for the fine-tuning process [50] because of its remarkable performance in Arabic NLP tasks [50]. This process is done using the Hugging Face Trainer API [51], with training and evaluation datasets taken from the existing dataset. With an eight batch size and a 2e-5 learning rate, the model trained during three epochs. Then, the dataset using the optimized model was reclassified using updated sentiment predictions.

C. Evaluation via Rating-Based Sentiment Mapping

The quality of the sentiment labels generated by BERT was compared to labels obtained from the numerical rating column. The same criteria as before were used to correlate ratings to sentiment (4–5 stars = positive, 3 stars = neutral, 1–2 stars = negative).

The comparison demonstrated an approximate agreement accuracy of 80.5% between the BERT-labeled sentiment and the rating-based sentiment, signifying a substantial alignment between textual sentiment and user ratings. This accuracy is deemed acceptable due to the subjective nature of feeling and the weak supervision characteristics of the task. To ensure the consistency and quality of the data, it's essential, prior to any analysis or implementing any ML models, to preprocess the dataset.

To ensure reliable results and enhance model performance, effective preprocessing was crucial. And this may include several cleaning procedures, such as normalization and textual data cleansing.

1. Text Cleaning

This cleaning process targets removing the noise and irrelevant details from the Arabic reviews, ensuring only important linguistic content is kept for further analysis [52-54]. The Python's re package for regular expressions was used in this process, which comprised the subsequent steps:

- URL Elimination: Regular expression patterns were used to eliminate all hyperlinks, such as those starting with www, http, or https. This step ensures getting rid of external web references that don't support SA.
- Filtering Emojis: A Unicode-based comprehensive regular expression was used to detect and filter emojis. This includes a variety of symbol sets, pictographs, and emoticons that can introduce bias and noise into the feature space
- Hashtags and Mentions Elimination: This step includes the elimination of the hashtags, such as #subject, and Twitter-style usernames like @user_name, since they are particular to a platform typically or don't have semantic meaning consistently in the context of course review sentiment.
- Filtering of Character: This process includes the filtering of numerals, Latin script, non-Arabic characters, and symbols to make the concentration of study on Arabic content. Regular expression was used to preserve Arabic characters where the Unicode range is \u0600-\u06FF, Arabic numbers where the Unicode range is \u0660-\u0669, and whitespace solely. This ensures that the processed content has linguistic integrity.
- Punctuation Removal: This step includes getting rid of a wide range of Latin and Arabic punctuation symbols, such as the question mark (؟), Arabic comma (،), and Arabic semicolon (؛).
- Normalization of Whitespace: This includes the removing of leading and trailing spaces and reducing consecutive whitespace characters into single spaces. And this leads to improving the consistency of the final text and facilitating the standardized token bounds.

The thorough cleaning method was particularly crucial for Arabic SA, as noise and script variations can greatly affect model performance. An example review before and after the cleaning process is shown below:

Before Text Cleaning	هذا الكورس رائع جداً! تعلمت الكثير، الـــرابط: https://example.com @user #تعليم"
After Text Cleaning	هذا الكورس رائع جدا تعلمت الكثير الرابط تعليم

2. Normalization

In this phase, the Arabic reviews went through normalization, which focuses on standardizing the characters and reducing orthographic diversity, leading to enhanced consistency and quality of subsequent tokenization and feature extraction.

The pipeline of this phase used the common utilities of the CAMeL Tools library with new regular expressions specifically designed for the Arabic content [55]. The following transformations were put into effect:

- Normalization of Unicode: A normalize_unicode function is used to standardize the Arabic characters by creating a single standard form from several Unicode representations of the same character, ensuring compatibility with various Arabic text sources.
- Removal of Diacritic: This aims to use dediac_ar from CAMeL Tools to make the elimination of Arabic diacritics (Tashkeel), meaning the short vowels, which are generally optional in written Arabic, leading to the lexical variation reduction and improved token matching.
- Normalization of Alef Variations: This means the use of normalize_alef_ar from CAMeL Tools to consolidate various representations of the Arabic letter Alef (e.g., "أ"،"إ"،"آ") into a standard "ا". This is important because, although they can cause inconsistencies during tokenization and vectorization, both variants are commonly used interchangeably in informal writing.
- Kaf Normalization: Various forms of the Arabic letter Kaf (e.g., "ک"،" ڪ"،" ک") were converted to the normal Arabic form "ك" through the use of a regular expression, reducing the discrepancies emerging from Persian, Urdu, or variations in style.
- Tatweel Removal: This means the elimination of the tatweel character (ـــ), utilized for visual elongation, using a regular expression (\u0640+), as they have no semantic value and disrupt token-level processing.
- Normalization of Whitespace: Unnecessary spaces produced during the cleaning or normalizing phases were eliminated, and the leading or trailing whitespace was gotten rid of.

The normalized form of the review text functions as the input for tokenization, segmentation, and subsequent morphological processing processes. Below is an example of a normalized review:

Before Normalization	"هذا الكورس رائـــع جداً تعلمت الكثير الرابط تعليم"
After Normalization	"هذا الكورس رائع جدا تعلمت الكثير الرابط تعليم"

3. Stopword Removal

This phase was implemented to further improve Arabic review texts by eliminating commonly used yet semantically weak phrases (functional words), such as prepositions, conjunctions, and articles, which have, in general, minimal semantic importance in sentiment classification tasks [56]. The process is done by creating a comprehensive Arabic stopword list that includes the default Arabic stopwords provided by the Natural Language Toolkit (NLTK) library and a custom-defined list of frequently found stopwords in the dataset.

And as semantics is important in the realm of SA, it was essential to hold onto negation terms, such as "ال"،"ما "،"ليس "", and "لـــم"", as they can reverse the meaning of neighboring sentiment-laden words [57]. Therefore, a carefully chosen list of negation terms is omitted from the stopword collection.

The definitive stopword list, excluding the negation phrases, was employed to filter the

normalized text. Each review was tokenized using whitespace, and all tokens that corresponded to stopwords or were fewer than two characters were eliminated.

Before Stopword	"هذا الكورس رائع جدا تعلمت الكثير الرابط تعليم"
After Stopword	"الكورس رائع جدا تعلمت الكثير الرابط تعليم"

4. Tokenization

In this phase, the text will be divided into smaller parts, such as words or phrases. In Arabic NLP [58], tokenization is considered a substantial challenge because of its morphological complexity, clitics, and affixes. A custom tokenization method was

devised to optimize processing efficiency and accuracy. The procedure included the following essential elements:

- Preservation of Sentiment Phrases: At this stage, a predetermined set of sentiment phrases (e.g., "جيد جدا"، "سيء للغاية"، "غير مفيـــد") was used to identify the concatenation of these multi-word expressions with underscores (e.g., "سيء للغاية" → "ســـيء_للغاية"), ensuring that their complete semantic value would remain intact after tokenization.
- Tokenization was conducted via the CAMeL Tools library [55], which comprised Basic Word Tokenizer (simple_word_tokenize) for preliminary segmentation and MorphologicalTokenizer integrated with a pretrained MLE disambiguator (calima-msa-r13) for enhanced analysis and precise segmentation according to the D3 tokenization framework. To enhance control and integration with sentiment phrases, the final implementation employed a whitespace-based tokenization method followed by post-processing.
- Substitution of Underscore: The temporary underscores were converted back to spaces, as these underscores were used to maintain the phrases.
- Removal of Non-informative Tokens: All empty or null tokens or non-informative morphological segments (such as affixes like "س"، "هـــم") generated during the preprocessing phases were discarded.

This approach facilitated the preservation of contextual sentiment expressions and avoided over-segmentation, a common problem in Arabic tokenization. Illustration of a customized tokenization example:

Before Tokenization	"الكورس سيء للغاية ولا أنصح به"
After Tokenization	['الكورس'، 'سيء للغاية'، 'ولا'، 'أنصح'، 'به']

5. Stemming and Lemmatization

At this phase, a selective stemming strategy was used to reduce morphological variation in Arabic words and improve the consistency of the tokenized evaluations. Due to the intricate derivational morphology of Arabic, which produces multiple surface forms for the same root meaning, stemming is crucial [59-61].

The process employed a selective integration of stemming and lemmatization as outlined below:

- The ISRI stemmer from NLTK was employed for most terms. It is specifically engineered for Arabic and eliminates prevalent affixes, including prefixes such as "ال" and suffixes like "ون", to distill words to their approximate root form. For instance, "يستخدمون" transforms to "خدم".
- Selective Lemmatization: A predetermined list of significant or commonly mis-stemmed terms (lemmatization_words) was established based on empirical observations. The terms in this list were lemmatized utilizing WordNetLemmatizer to maintain their accurate semantic forms instead of diminishing them to abstract roots. This hybrid methodology mitigated the distortion of sentiment-laden terms.

Example of Selective Stemming

Before Stemming and Lemmatization	['الكورس'، 'سيء للغاية'، 'ولا'، 'أنصح'، 'به']
After Stemming and Lemmatization	['كورس'، 'سيء للغاية'، 'ولا'، 'نصح'، 'به']

2.2 Feature Extraction

Two techniques are used for textual representation: TF-IDF captures term importance in context, and FastText provides semantic word embeddings, useful for understanding word meanings in different contexts [62-64].

2.2.1 TF-IDF Representation

TF-IDF is a classic statistical technique used to show the importance of a word within a document in a corpus. It takes into consideration both the frequency of a word within a particular review (term frequency) and its rarity across all reviews (inverse document frequency), so emphasizing terms that are unique and informative.

A TfidfVectorizer from the scikit-learn library was utilized; the parameters for the TF-IDF vectorizer were determined through practical testing and established best practices in Arabic SA.

The ngram_range =(1, 2) was selected to encompass both unigrams (individual words) and bigrams (two-word combinations), as sentiment in Arabic is frequently conveyed through concise phrases (e.g., "سيء للغاية"،"جيد جـــدا"). At the phrase level, the integration of bigrams enhances the model's ability to understand sentiment and context.

The parameter max_features = 5000 was employed to restrict the feature space to the 5,000 most prevalent and significant n-grams. This diminishes dimensionality, decreases memory consumption, and alleviates overfitting—particularly crucial due to the dataset's very limited size. A grid search was performed to assess various values: 3000, 5000, and 7000, by comparing the performance of a baseline classifier (SVM) using accuracy, precision, recall, and F1-score metrics.

Augmenting the max_features parameter in TF-IDF enables the model to save a greater number of n-grams, thereby enhancing the vocabulary coverage, particularly for infrequent or more specific sentiment-laden expressions. Nonetheless, this expanded coverage entails trade-offs: it introduces additional sparse features and may incorporate noisy or irrelevant terms, thus diminishing model generality. In our experiments, augmenting the number of features from 3000 to 7000 yielded negligible increases in performance. The configuration with 5000 characteristics attained a commendable equilibrium, with an accuracy of 85.25% and an F1-score of 85%, signifying good precision and recall across classes. Despite the 7000-feature configuration achieving marginally superior accuracy (85.94%) and recall (86%), the enhancement was negligible and incurred increased sparsity and computing demands. Consequently, 5000 was chosen as the ideal value for max_features, providing the most favorable balance of model performance, vocabulary coverage, and efficiency. The resultant TF-IDF matrix was sparse and high-dimensional, ideal for models that operate efficiently with linear feature representations.

Table 2.1

Performance metrics for various max_features values utilizing 5-fold cross-validation

Max Features	Accuracy	Precision	Recall	F1-score
3000	84.48%	84%	84%	84%
5000	85.25%	85%	85%	85%
7000	85.94%	85%	86%	85%

2.2.2 FastText Embeddings

FastText word embeddings are used to explain the deep semantic and contextual links between words in Arabic text. It's created by Facebook AI Research and represents words as collections of character n-grams and can handle out-of-vocabulary (OOV) terms and word form changes, making it proficient for Arabic, a morphologically rich language.

The 300-dimensional Arabic embedding (cc.ar.300.bin), which was learned on Common Crawl data, was used as the pre-trained FastText model, loaded through the FastText Python library. For each review, a fixed-length dense vector representation was produced by taking out and averaging the embeddings of each individual token. In cases where the token was absent from the FastText vocabulary, it was either left out or given a zero vector. This represents the model's ability to understand semantic similarities and manage OOV terms and assist more intricate classification structures.

2.2.3 Integration of TF-IDF and FastText

The desired hybrid feature representation was done by the integration of TF-IDF features with FastText to make use of both local statistical significance and global semantic relevance, leveraging the advantages of both methodologies. As the TF-IDF measures the relative importance of words and phrases (bigrams and unigrams) within the corpus, it highlights instructive terms. And embedding provides a semantically improved dense representation of words, encompassing uncommon or morphologically complex terms by representing words as combinations of subword character n-grams. The amalgamation procedure encompassed the subsequent stages:

1. The TF-IDF matrix was converted into a sparse matrix. Then, it was normalized using L2 normalization. To make sure that the length of all feature vectors had one unit.
2. The FastText sentence vectors, derived by averaging the word embeddings for each review, were converted into a NumPy array. Then it was normalized using the L2 norm. This reduces the potential that any individual vector will overshadow others because of differences in magnitude.
3. The normalized matrices were further converted into sparse format and horizontally concatenated using hstack, yielding a final composite feature matrix that preserves both feature types.

The resultant feature matrix exhibited a dimension of (n_samples, 5300), with the initial 5000 columns corresponding to TF-IDF features and the subsequent 300 columns denoting the FastText embeddings.

The integrated representation enabled the sentiment classification model to concurrently leverage both term-level significance and semantic meaning, enhancing its capacity to manage intricate expressions and subtle opinions in Arabic course evaluations.

2.3 Sentiment Analysis

A sentiment analysis module is applied to generate two key outputs, leading to the facilitation of the integration of user sentiment into the RS, hence enhancing its personalization and relevancy. The first output is categorical sentiment classifications, assigning each review as positive, neutral, or negative. The second output is a sentiment score, a numerical value representing the polarity of the text.

The sentiment scores were first organized chronologically to represent a user's sentiment over time evolution. These scores were then exposed to a simple linear regression, where sentiment values were the output and time steps were the input. The resulting line's slope presents the sentiment trend; a positive slope refers to feedback that is getting more positive over time, while a negative slope refers to sentiment that is declining over time.

2.3.1 Selection of Classification Model

For the sentiment classification challenge, various SL ML classifiers were used, including RS, gradient boosting (GB), and SVM. All of these models were trained and evaluated using the identical composite feature representation (TF-IDF and FastText). This way ensures an even comparison and depends only on the classifier's ability to learn from the data, without any discrepancies in the representation of the input.

Five-fold cross-validation was used to evaluate each of these three classifiers using accuracy, precision, recall, and F1-score metrics. It was found that SVM in all criteria exceeded the other two models consistently. The other two models show a satisfactory performance, but they are less efficient in handling the high-dimensional and sparse feature vectors generated by the hybrid representation.

Table 2.4 illustrates the comparison between these three classifiers. The SVM achieved the highest test accuracy (85.26%) and macro-average F1-score (85%), although the other two classifiers exhibited satisfactory performance. This means that SVM is the most effective to be used for this task by demonstrating consistent and fair performance across all sentiment categories. Additionally, SVM has shown a reduced overfitting to GB, which has significantly shown greater training accuracy (95.35%) but markedly lower testing accuracy (75.68%). Based on what was found, SVM was chosen as the definitive model for sentiment categorization.

Table 2.2

Evaluation of Classifiers' Performance Utilizing Integrated TF-IDF and FastText Features

Model	Train Accuracy	Test Accuracy	Precision (Macro)	Recall (Macro)	F1-Score (Macro)
RF	87.81%	66.94%	65%	65%	65%
GB	95.35%	75.68%	74%	74%	74%
SVM	94.43%	85.26%	85%	85%	85%

2.3.2 Model Training and Cross-Validation

Upon picking SVM as the definitive classifier, the model underwent training and evaluation through a systematic cross-validation method to guarantee the reliability and generalizability of the outcomes.

a. Dataset Preparation

The dataset in this study consists of a set of reviews related to Arabic language courses, which were classified as positive, neutral, and negative. A fine-tuned Arabic BERT model intended to classify review sentiment based on the semantic content of the preprocessed Arabic text was used to generate the sentiment labels automatically.

Initial examinations and analyses regarding the distribution of categories showed a notable imbalance, as it was noted that the positive ratings were more frequent than the neutral and negative. This discrepancy ran the danger of the existence of bias of the classifier towards the majority class, which may lead to reducing its ability to classify underrepresented sentiment types in an accurate manner.

In order to solve this issue, a random upsampling strategy was used on the training data to equilibrate the amount of instances across each sentiment category. The reviews from

minority classes (neutral and negative) were taken and then randomly replicated until all of these three classes were represented in the training set equally. This strategy of upsampling was applied solely on the training subset while preserving the original distribution on the validation and test sets, similar to the approaches used in [65-66].

This strategy leads to the needed balance that ensures that the model will expose an equivalent quantity of training data for each sentiment class, hence cultivating equitable learning and enhanced generalization across all classes.

After the dataset became balanced, the SVM model was trained with a radial basis function (RBF) kernel via the hybrid feature representation. SVM was chosen for its efficacy in managing high-dimensional, sparse feature spaces, common in tasks involving NLP, particularly when integrating TF-IDF with embeddings. The feature representation, which comprised

- TF-IDF vectors utilizing the foremost 5000 n-grams (unigrams and bigrams).
- FastText embeddings, in which each review was represented by the mean of 300-dimensional word vectors.
- The final input matrix was a 5300-dimensional vector for each review, created by horizontally concatenating the TF-IDF and FastText vectors.

The SVM model was executed utilizing SVC from the scikit-learn library with the specified hyperparameters:

- C: Magnitude of regularization
- gamma: Kernel coefficient for the RBF
- kernel: Facilitates non-linear categorization boundaries
- class_weight: Modifies weights inversely according to class frequencies.
- probability: Activates probability estimations for ROC analysis
- random_state: Guarantees repeatability

A grid search through a range of possible parameter values was used to specify the values, employing cross-validation inside the training dataset, leading across all sentiment categories to optimal generalization performance.

The dataset was divided into 80% for training and 20% for testing by a normal random split, utilizing a fixed random seed to ensure reproducibility. Although the split was not

stratified, the data had been balanced through upsampling, ensuring about equal representation of all sentiment classes throughout the divides.

The SVM model went through training on the training set through 5-fold cross-validation.

The training data was split into five equal folds. In each iteration, four folds were used for training, while one was used for testing. This process was repeated five times, allowing each fold to serve once as a testing set. The performance metrics were documented and averaged across all folds from each iteration. This cross-validation process was repeated for each hyperparameter combination, and the combination that got the highest performance was chosen as the optimal configuration for SVM.

This way of training was used in order to make the reduction in the risk of overfitting in addition to ensuring the model generalization effectively to novel data.

b. Conclusive Model Assessment

The chosen sentiment classification model was an SVM with an RBF kernel, trained on the upsampled dataset utilizing a mixed feature representation of TF-IDF and FastText embeddings. The model was assessed using a reserved test set (20% of the data) that was excluded from training and cross-validation to guarantee an impartial evaluation of generalization performance.

The optimal model configuration employed the subsequent hyperparameters:

- kernel = 'rbf'
- C = 7
- gamma = 0.1
- class_weight = 'balanced'
- probability = True
- random_state = 42

The model attained robust and equitable outcomes across all mood categories. The principal evaluation metrics for the test set are outlined below:

Table 2.3

Test Set Performance

Metric	Score
Accuracy	85.26%
Precision (macro)	85%
Recall (macro)	85%
F1-score (macro)	85%

2.3.3 Evaluation Metrics

Various evaluation metrics and diagnostic tools were utilized to evaluate the performance and reliability of the sentiment categorization model [67-70]. These metrics offer numeric performance scores and visual representations of the model's behavior across the three sentiment categories: positive, neutral, and negative.

A classification report is a performance evaluation metric utilized in machine learning to evaluate the efficacy of classification models. It offers essential indicators for each class in the dataset, facilitating an assessment of the model's performance. It encompasses the subsequent metrics for each category:

a. Precision: The ratio of accurately predicted instances to the total expected instances for a specific class.

$$Precision = \frac{TP}{(TP+FP)} \text{ ……..…….. (2.1)}$$

b. Recall: The ratio of accurately anticipated instances to the total actual instances of that class.

$$Recall = \frac{TP}{(TP+FN)} \text{ ……....…….. (2.2)}$$

c. F1-Score: The harmonic mean of precision and recall, offering a balanced metric notwithstanding class imbalance.

$$F1\ score = 2\ \frac{(Precision*Recall)}{(Recall+Precision)} \text{ …….. (2.3)}$$

d. Accuracy: The aggregate percentage of accurately categorized samples across all classifications.

$$Accuracy = \frac{(TP+TN)}{(TP+FP+TN+FN)} \text{ ……..... (2.4)}$$

Where TP is instances in which the model accurately identified a positive category, TN is instances where a model correctly identified the contrary category, FP is instances where the model erroneously classified the positive category, and FN is instances where the model erroneously classified the negative category.

Alongside per-class scores, both macro-averaged and weighted-averaged metrics of precision, recall, and F1-score were employed to evaluate overall model performance. These averages address class imbalance and offer a comprehensive assessment of categorization efficacy.

A confusion matrix is a tabular representation utilized to assess the efficacy of a classification model, particularly in binary and multiclass classification scenarios. And this shows the extent to which the actual labels and the model's predictions align. It made it easier to identify prevalent misclassification patterns and evaluate the relative difficulty of differentiating specific classes.

The ROC curve is a graphical tool used for efficacy evaluation for the binary classification model. It shows across different threshold levels the true positive rate (TPR) in opposition to the false positive rate (FPR) across different threshold levels. The Area Under the Curve (AUC) was calculated to capture the efficacy of the model in differentiating across groups.

$$FPR = \frac{FP}{(FP+TN)} \text{ …….. (2.5)}$$

$$TPR = \frac{TP}{(TP+FN)} \text{ . …….. (2.6)}$$

The precision-recall curve (PR) is a graphical representation used across different classification thresholds for the performance evaluation. It focuses on the balance between recall and precision without any effect from true negatives, which is beneficial.

A learning curve is a graphical tool used in ML for the evaluation of model performance as the volume of training data increases. This curve shows the variation in the accuracy or errors of the model while training with the volume of data increasing, which makes it easier for comprehension, overfitting, and underfitting.

2.4 Recommendation Model

This section delineates the process for constructing the course RS utilizing CF through implicit matrix factorization. The system utilized the ALS method from the implicit Python library. Two configurations were analyzed: one incorporating sentiment integration (RS+SA) and the other excluding it (RS-SA).

The core recommendation model uses a hybrid approach, CF, that provides item suggestions based on user similarity, and a sentiment-aware interaction score is fused into the model to improve accuracy.

Two user categories are handled differently:

- Cold User Model (K-Means & KNN): For users with limited data, clustering and similarity-based approaches are used.
- Active User Model (ALS): For users with sufficient history, ALS matrix factorization is applied.

2.4.1 Problem Formulation

The RS aims to predict a student's propensity towards a course based on prior interactions. In this study, unlike traditional ratings-based systems, interactions will be relied upon as implicit feedback inferred from observed behavior rather than just from explicit ratings.

Every interaction between a student and a course is quantified by an interaction score, encompassing engagement metrics such as ratings and SA results.

2.4.2 Data Representation

To do the training of the recommendation model using matrix factorization, a user-item interaction matrix was built, where rows refer to distinct students, columns refer to distinct courses, and cell values refer to the interaction ratings between students and courses.

- A unique identifier was provided to each student based on their review history.
- Courses were distinctly distinguished by their course ID.
- The dataset comprised around 10,654 students and 570 courses.

This work integrates SA findings into the development of the user–item interaction matrix to enhance user preference signals and elevate recommendation quality. Rather than depending exclusively on numeric ratings, sentiment scores derived from user evaluations were incorporated as supplementary indicators of customer satisfaction.

Two sentiment-related attributes were calculated for each review:

Sentiment Score: A continuous value indicating the polarity of the user's review, derived using a fine-tuned Arabic sentiment classification model. The score was standardized on a scale from 0 to 1, representing the emotional tone of the evaluation as favorable, neutral, or negative.

Sentiment Trend: A customized metric reflecting the progression of a user's sentiment over time. A linear regression was conducted for each user on their chronological sequence of sentiment scores to determine the slope of sentiment change. An upward trend signifies enhanced satisfaction, whereas a downward trend denotes diminishing involvement.

The sentiment signals were regarded as implicit feedback indicators and incorporated into the comprehensive interaction score utilized for training the recommendation algorithm.

To assess the impact of sentiment on suggestion efficacy, two configurations of the RS were examined:

- RS-SA (RS Excluding SA): Interaction score calculated solely based on rating and behavioral attributes (e.g., rating, recency, popularity).
- RS+SA (RS Including SA): The interaction score incorporated sentiment score and sentiment trend as supplementary weighted elements.

This design facilitates the isolation of the effects of sentiment integration through a controlled comparison. Therefore, by contrasting these two versions of models, the study aimed to determine whether sentiment-enhanced signals may produce more

accurate and tailored course recommendations, particularly in situations with limited data.

A composite interaction score was created to capture user preferences beyond basic ratings. This score amalgamates behavioral, environmental, and sentiment-driven information to function as the implicit feedback input for matrix factorization.

The interaction score consolidates multiple factors that reflect user participation, course popularity, recency, and sentiment. Each feature is normalized or transformed as necessary and assigned a weight to indicate its importance in the final score. The resultant raw interaction ratings were further standardized using Min-Max scaling to conform to the range [0, 1].

This composite interaction score allowed the model to transcend conventional single-signal feedback (e.g., rating-only) by integrating sentiment, temporal dynamics, course features, and behavioral inputs. Consequently, the RS was enhanced to assimilate nuanced, implicit user preferences, thereby producing more tailored and precise course choices. The score was subsequently utilized as the input for training the matrix factorization model via ALS.

The comprehensive interaction score for user u and item i is represented as a weighted linear amalgamation of conventional CF indicators and sentiment-informed attributes:

$$I(u,i) = \theta(u,i) + w_s.S(u,i) + w_t.T(u) \quad \text{..... (2.7)}$$

Where:

- $I(u,i)$: aggregate interaction score
- $\theta(u,i)$: aggregation of non-sentiment features
- $S(u,i)$: sentiment score
- $T(u)$: sentiment trend for user u
- w_s,w_t: weights for sentiment score and trend

The non-sentiment element is articulated as:

$$\theta(u,i) = w_r.R(u,i) + w_d.exp\big(-\alpha.d(u,i)\big) + w_l.L(u,i) +$$

$$w_{uc}.min\left(\frac{n_u}{n_i},1\right) + w_a.R(i) + w_e.C(i) + w_f.F(u,i) \quad \text{..... (2.8)}$$

Where:

- $R(u,i)$: normalized rating
- $exp(-\alpha.d(u,i))$: recency decay
- $d(u,i)$: days since review
- $L(u,i)$: review length
- n_u,n_i: quantity of evaluations by the user and pertaining to the course
- $R(i)$: average course rating
- $C(i)$: category weight
- $F(u,i)$: interaction feedback signal
- α: temporal decay constant
- w_*: tunable weights

These hyperparameters (weights) were iteratively tested and empirically optimized. The adjustment led to the optimization of suggestion efficacy by balancing the effects of the traditional behavioral indicators and sentiment-augmented input. All input features were standardized with min-max scaling to guarantee equitable contribution to the final interaction score.

2.4.3 Collaborative Filtering Approach

In this system, the ALS matrix factorization using the implicit Python library was used to generate tailored course recommendations [72-74]. ALS is considered a popular and scalable CF method, as it works for implicit feedback data effectively, where either there are no direct ratings or they are incorrect.

1. Why ALS?

The recommendation task in this study depends on the implicit input; instead of determining user preferences explicitly, it is deduced. In these contexts, conventional CF techniques, such as user-based or item-based kNN, frequently prove inappropriate because of data sparsity and the lack of dependable negative feedback.

Therefore, to address these issues, ALS was used. It is especially distinguished by efficiency and skill at handling implicit data. As it converts observed interactions into confidence levels and treats absent interactions as possible negative signals with less weight.

The fundamental concept of ALS is to decompose the user-item interaction matrix R into two low-dimensional matrices:

$$R \approx X.Y^T \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (2.9)$$

Where:

- $X \epsilon R^{u \times f}$: user latent feature matrix
- $Y \epsilon R^{i \times f}$: item (course) latent feature matrix
- f: count of latent factors

Rather than explicitly forecasting ratings, ALS optimizes the subsequent confidence-weighted loss function:

$$\sum_{u,i} \quad c_{ui}.(p_{ui} - x_u^T y_i)^2 + \lambda \left(||x_u||^2 + ||y_i||^2 \right) \ldots\ldots\ldots\ldots\ldots\ldots \quad (2.10)$$

Where:

- $p_{ui} \epsilon \{0,1\}$: binary preference (1 indicates the presence of interaction, 0 indicates its absence)
- $c_{ui} = 1 + \alpha.r_{ui}$: confidence score
- λ: Regularization parameter to prevent overfitting
- $x_u \; and \; y_i$: latent vectors for users and items

This study utilized the default formulation in the implicit ALS implementation, where confidence is directly obtained from the normalized interaction scores, in contrast to methods necessitating confidence scaling with a manually specified α. This enabled the model to seamlessly include both robust and feeble engagement signals without supplementary scaling factors.

With this formulation, ALS can use matrix factorization to scale to sparse, extensive datasets efficiently, handle absent interactions effectively by attributing to them minimal confidence levels, and focus on robust user-item preferences from interaction ratings that were recorded.

ALS was a logical choice because this study employed a composite interaction score. The model is particularly good at simulating complex, sentiment-augmented

engagement signals in educational content because of its confidence-based methodology, which allows it to discriminate between robust and weak interactions.

2. Model Configuration

The model was constructed with hyperparameters that were experimentally optimized to achieve a balance between accuracy and generalization.

To reach the optimal ALS setup, a randomized search using ParameterSampler from scikit-learn was performed across a predetermined hyperparameter space. For every sample combination, the model was trained on the interaction matrix and then evaluated using RMSE on a reserved test set. The computation of RMSE is done by comparing actual user preferences with the expected top-N course scores, concentrating only on the items that overlapped between the forecasts and the ground truth. After ten setups were evaluated, the one with the lowest average RMSE was chosen.

The identical ALS hyperparameter configuration—150 latent components, regularization set to 1.0, and 10 iterations—demonstrated optimal performance for both RS+SA and RS-SA. This indicates that the overall structure and scale of the interaction matrix were consistent across both configurations and that the incorporation of sentiment data had a greater impact on the recommendation output than the optimal model capacity or regularization strength.

The model was trained using the user–item matrix derived from the calculated interaction score, enabling the ALS algorithm to learn from the intensity of engagement rather than solely from binary preferences.

3. Training Process

Prior to training:

- The interaction matrix was transformed into a sparse matrix structure to save memory consumption and enhance computational speed.
- ALS was trained using the implicit feedback objective, wherein unseen interactions are regarded as potential negatives with inherent low confidence. No supplementary confidence scaling, such as an alpha parameter, was utilized; rather, confidence was calculated internally based on the normalized interaction score.

Subsequent to training:

- The model generated top-N course recommendations for each user by calculating the dot product of user and item embeddings.
- Performance was assessed utilizing measures appropriate for implicit feedback, as outlined in the subsequent section

2.4.4 Handling Cold Start Users

CF algorithms, such as ALS, depend significantly on user-item interaction history to derive meaningful representations. In practical e-learning settings, it is prevalent to encounter cold-start users—learners who have provided few reviews or ratings. Cold-start users in this study are those with less than three interactions. A specialized recommendation pipeline was developed to address this issue, including user clustering, feature-based similarity, and hybrid kNN methodologies [75-77].

2.4.4.1 Cold-Start User Profiling and Clustering

A feature-based profiling approach was employed to create user representations appropriate for cold-start recommendations. Every user was characterized by a collection of behavioral, contextual, and sentiment-sensitive attributes.

A StandardScaler was used for normalization of all numerical features using z-score transformation, which is an important step because the K-means clustering approach is considered distance-based and sensitive to the scale of features. Feature attributes such as user_review_count (can reach into the hundreds) would overshadow smaller-scale features such as recency_decay or sentiment_score. Consequently, only scaled numeric variables were utilized in clustering, whilst unscaled metadata (e.g., course name, course ID, URL) was omitted from the clustering process and employed subsequently for mapping and interpretation.

Using K-means, the users were categorized into two clusters, enabling the system to specify latent behavioral similarities among users based on their limited profile data. The decision to choose the cluster count (K=2) goes back to the Silhouette Score tool [78], which is usually used for the evaluation of clustering, which assesses how closely each data point matches its assigned cluster relative to other clusters. Among several

evaluated values of K (ranging from 2 to 10), the setup with two produced the highest silhouette score, indicating a clear division between two separate user groups.

Based on the model configuration, different features were used for clustering. In RS-SA, the grouping used only behavioral and temporal features such as user_course_ratio, recency_decay, interaction_score, days_since_review, avg_course_rating, and user_review_count. In contrast, in RS+SA, further features were incorporated, such as sentiment_score and sentiment_trend.

This approach allowed the RS+SA model to create more comprehensive user profiles that incorporated both behavioral patterns and emotional signals, facilitating more nuanced cold-start recommendations.

2.4.4.2 Hybrid Recommendation Strategy

A. User-Based kNN within Cluster

For each cold-start user, the system initially established the cluster assignment based on their standardized behavioral and contextual attributes, employing the previously disclosed K-Means model. To facilitate significant similarity assessments, only users inside the identical cluster were regarded as prospective neighbors.

The system calculated the cosine similarity between the feature vector of the cold-start user and those of all other users within the same cluster. These features encompassed (contingent upon RS-SA or RS+SA configuration):

- user_course_ratio
- recency_decay
- days_since_review
- user_review_count
- avg_course_rating
- interaction_score
- RS+SA (+ sentiment_score, and sentiment_trend)

The top-N most analogous users, determined by the highest cosine similarity, were chosen as nearest neighbors. The system recognized and ranked routes based on the frequency of interactions with these neighbors. The courses with the highest interaction rates were chosen as recommendations for the cold-start user.

$$Recommended_u^{user-based} =$$

$$Top_K\ courses\ by\ frequency\ from\ top_N\ similar\ users \quad \text{..... (2.11)}$$

The user-based kNN component offered behaviorally informed course recommendations by drawing parallels with other users possessing more extensive histories, facilitating individualized suggestions despite the target user's limited engagement.

B. Item-Based kNN on Course Names:

To enhance user-based suggestions, an item-based kNN method was employed to suggest courses that are textually analogous to those previously engaged with by the cold-start user. This component is especially beneficial when a cold-start user has evaluated only one course, as it enables the system to extrapolate recommendations based on the semantics of course material.

Vectorizing Course Names: All distinct course names inside the user's designated cluster were retrieved and converted via TF-IDF vectorization. The TfidfVectorizer quantified the significance of terms in course titles, allowing the system to evaluate courses based on their textual descriptions.

Assume:

- $C = \{c_1, c_2, \ldots, c_n\}$: constitute the collection of all course titles within the cluster
- V_i: TF-IDF vector for course c_i

The outcome was a vector space representation of all cluster courses.

Encoding the Cold User's Known Course: The course title previously interacted with by the cold-start user was vectorized using the identical TF-IDF model, resulting in a query vector.

$$Q = TFIDF(user`s\ course\ name) \quad \text{..... (2.12)}$$

Similarity Calculation and Ranking: Cosine similarity was calculated between the user's course vector Q and each course vector V_i in the cluster.

$$Sim(Q, V_i) = \frac{Q.V_i}{||Q||.||V_i||} \quad \text{..... (2.13)}$$

The most similar top-K courses were chosen according to these similarity scores. These items functioned as content-based suggestions, indicating semantic similarities in course titles and thematic content.

C. Recommendation Integration

The last list of recommendations was produced by combining the outcomes from user-based and item-based kNN models into a singular ranked list. The proposed courses were unified by both methodologies, and superfluous items were also acknowledged and evaluated based on their frequency among the sources. This score system highlighted courses in both user and item recommendation watercourses, hence improving pertinence through cross-model concordance. The consolidated recommendation list was systematized later by score, and the top-K most widespread courses were selected as the final suggestions. This approach of integration coordinated between the behavioral similarity with content-based pertinence permits the system to supply more diverse and exact recommendations to cold-start users with minimalistic interaction histories.

2.5 Evaluation

The system is evaluated using standard metrics such as Precision@K, Recall@K, or RMSE to measure the effectiveness of incorporating sentiment into the recommendation pipeline [79-85]. The following is a comprehensive elucidation of each metric employed.

RMSE quantifies the mean squared deviation between expected and actual interaction scores.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \quad (r_i\hat{} - r_i)^2} \qquad \text{..... (2.14)}$$

Where:

- $r_i\hat{}$: predicted score
- r_i: true interaction score
- n: aggregate count of assessed interactions

Precision@K quantifies the ratio of pertinent things within the top-K recommendations that the user has engaged with.

$$Precision@k = \frac{1}{|U|}\sum_{u\epsilon U} \quad \left(\frac{|Recommended_u^K \cap Relevant_u|}{K}\right) \text{ …….. (2.15)}$$

Where:

- $Recommended_u^K$: top-K recommended items for user u
- $Relevant_u$: items with which user u engaged in the test set
- K: count of recommended items
- $|U|$: Total user count

Recall@K quantifies the ratio of pertinent things effectively recommended inside the top-K list.

$$Recall@k = \frac{1}{|U|}\sum_{u\epsilon U} \quad \left(\frac{|Recommended_u^K \cap Relevant_u|}{Relevant_u}\right) \text{ …….. (2.16)}$$

NDCG@K assesses not only the recommendation of pertinent items but also their ranking within the list. A logarithmic penalty is imposed on lower-ranked results.

$$NDCG@K = \frac{1}{|U|}\sum_{u\epsilon U} \quad \frac{DCG@k_u}{IDCG@K_u} \text{ …….. (2.17)}$$

Where:

- $DCG@K = \sum_{i=1}^{K} \quad \frac{rel_i}{log_2(i+1)}$
- rel_i: relevance score (often 1 for relevant, 0 for non-relevant)
- $IDCG@K$: Optimal DCG for user u, utilized for normalizing

The success rate quantifies the percentage of users who received at least one pertinent item inside their top-K suggestions.

$$SuccessRate@k = \frac{1}{|U|}\sum_{u\epsilon U} \quad 1(|Recommended_u^K \cap Relevant_u| > 0) \text{ ……. (2.18)}$$

Where $1(\cdot)$ is an indicator function that yields 1 if the condition holds true and 0 otherwise.

Coverage quantifies the proportion of distinct things present in the top-K suggestions for all users.

$$Coverage@k = \frac{|U_u Recommended_u^K|}{|I|} \quad \text{..... (2.19)}$$

Where I is the total count of distinct objects within the collection.

Diversity assesses the degree of dissimilarity among the items in the recommendation list. It is frequently calculated as the mean pairwise dissimilarity among the top-*K*

recommended items for a given user:

$$Diversity@k = \frac{2}{K(K-1)} \sum_{i=1}^{K-1} \quad \sum_{j=i+1}^{K} \quad \left(1 - sim(i_u, j_u)\right) \text{..... (2.20)}$$

Where K is the number of recommended items and $sim(i_u, j_u)$ is the similarity between items i and j recommended to user u, typically computed based on item content or embeddings. The overall diversity across all users is given by:

$$Diversity@K = \frac{1}{|U|} \sum_{u \epsilon U} \quad Diversity@K_u \text{..... (2.21)}$$

For active users (those with previous interactions), the evaluation concentrated on Precision@K, Recall@K, NDCG@K, and RMSE. These metrics evaluate how well the algorithm categorized and rated content that users might interact with (accuracy).

For cold-start users (users with minimal or no prior encounters), evaluation focused on success rate, coverage, diversity, precision@K, and recall@K, which means the system's capacity to provide relevant, discoverable, and comprehensive recommendations to new users with limited profile information.

Chapter Three

Results and Discussions

In this chapter, a comprehensive analysis of outcomes that resulted from both SA and RS stages. Also, the system efficiency for both active and cold user scenarios. Moreover, a comprehensive comparison between the two systems (RS-SA and RS+SA) focuses on how sentiment integration influences the personalization and recommendation quality.

It is crucial to note that the results presented in this chapter are based originally on an English dataset and then translated into the Arabic language using AWS Translate. Despite AWS being a neural machine translation service that provides high-quality translation, there are limitations, such as capturing the full emotional depth and nuances embedded in natural language. Sentiment expressions and emotions usually differ across languages, and minor variations in context, tone, or cultural meaning during translation may not be fully preserved. Consequently, some loss of sentiment-related information may occur, which may affect the interpretation and performance of the SA and recommendation outcomes presented in this study.

3.1 Sentiment Analysis Results

The SVM classifier was trained to classify the sentiments using a hybrid feature representation that integrates TF-IDF and FastText embeddings. The dataset originally showed a class imbalance, with 8,600 positive, 5,370 neutral, and 6,026 negative reviews. To reduce the possible bias and improve model fairness across sentiment classes, a random upsampling strategy was applied only to the training set, ensuring a balanced distribution across all three sentiment categories. The upsampling was not applied to the test set to keep its real-world distribution, enabling a realistic evaluation.

The dataset was divided into training and test sets using an 80/20 random split, where 80% of the data was used for training and 20% for testing. The way of splitting was done using the train_test_split() function with a fixed random_state=42 to guarantee reproducibility.

The test set consisted of 4,131 reviews, comprised of 1,715 positive (2), 1,238 neutral (1), and 1,178 negative (0) reviews. It achieved 94.43% as training accuracy and

85.26% as test accuracy, showing no discernible overfitting with a strong generalization capability. The ~9% drop between training and test accuracy is expected and reasonable in ML and reflects that the model performs well on both seen and unseen data, indicating that the model picked up generalizable patterns in a successful manner and did not overfit the training data. Table 3.1 represents the classification report, showing the performance per class.

Table 3.1

The results of the SVM model classification performance on the test set are documented per class in terms of precision, recall, F1-score, and support (number of instances). The macro-average of 0.85 contemplates a balanced performance across all classes

Class	Precision	Recall	F1-score	Support
Positive	0.92	0.87	0.90	1715
Neutral	0.77	0.82	0.79	1238
Negative	0.84	0.87	0.85	1178
Macro Avg	0.85	0.85	0.85	4131

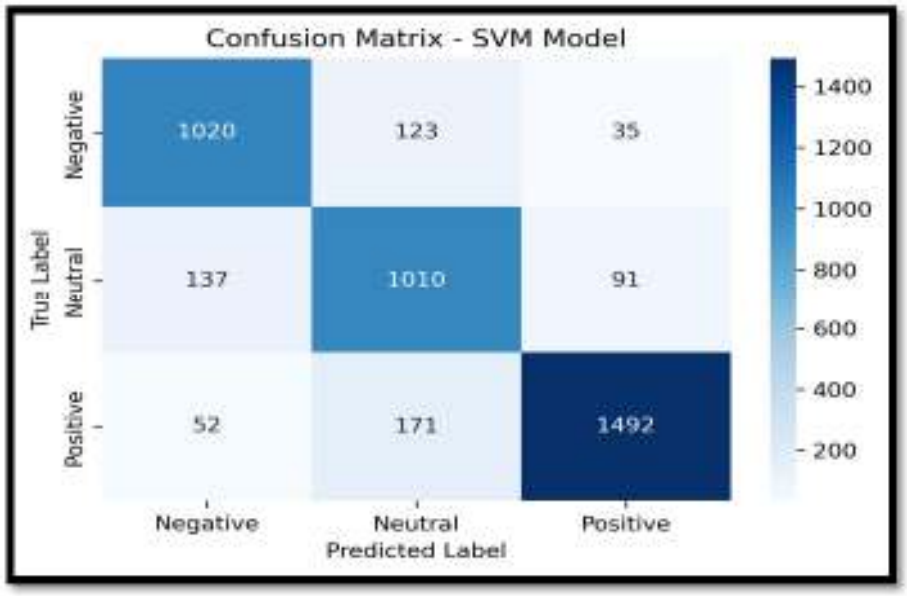
Figure 3.1 illustrates the confusion matrix of the SVM model. The majority of misclassifications occurred between the negative and neutral classes, showing a semantic overlap in student reviews that makes the differentiation between them more complicated. Positive sentiments were identified with 87% recall and negligible confusion.

This confusion between negative and neutral classes has been talked about before in related previous studies. The authors in [86-87] noted in their study that neutral sentiments intersect both linguistically and semantically with mild negativity, especially in short or informal texts, resulting in frequent misclassifications. Furthermore, these studies highlighted that the indirect or soft expression styles, ambiguous phrasing, and indirect criticism lead to ambiguity in sentiment classification.

Each cell in the confusion matrix represents the number of predictions generated for a certain pair of true and predicted labels. The diagonal cells with a dark blue color represent correctly classified occurrences. The off-diagonal cells with a lighter blue color represent misclassification occurrences. Darker shades indicate a greater density of samples, but the lighter shades indicate a lower density of samples.

Figure 3.1

Confusion matrix analysis



The SVM model achieved 1492 correct predictions for positive, 1010 for neutral, and 1020 for negative. Despite that, 11% of neutral reviews were categorized as negative, and 10% of negative reviews were categorized as neutral. This confusion and ambiguity is expected in sentiment analysis, particularly in authentic Arabic language, where moderately negative reviews often utilize neutral language and neutral reviews may have negative phrasing devoid of intense emotional cues.

The ROC curves depicting the TPR against the FPR for each sentiment class are shown in Figure 3.2. The AUC ratings represent the model's ability to differentiate between classes. The model shows high ability for discrimination, with AUC values of 0.97 for both positive (class 2) and negative (class 0) and 0.94 for neutral (class 1). These values are high AUC scores, which reflect the strong ability of the model to distinguish between each sentiment class from the others.

Although Class 1 shows a marginally lower AUC than the other two classes, which is natural and expected as neutral sentiments often overlap with linguistic patterns of both negative and positive sentiments and expressions, the model preserves strong classification performance across all classes. As for the curves near the upper left corner, which show the low false positive rate and high sensitivity achieved by the classifier, these results confirm the effectiveness of the SVM classifier and the extracted features' quality (TF-IDF and FastText) used in training.

The values as shown in Table 3.2 represent superb separability, especially for the positive and negative classes. The value for the neutral class is comparatively lower than other classes, which mirrors the ambiguity of that class, congruent with the confusion matrix analysis.

Table 3.2

AUC ratings for each sentiment class were derived from the ROC analysis of the SVM model

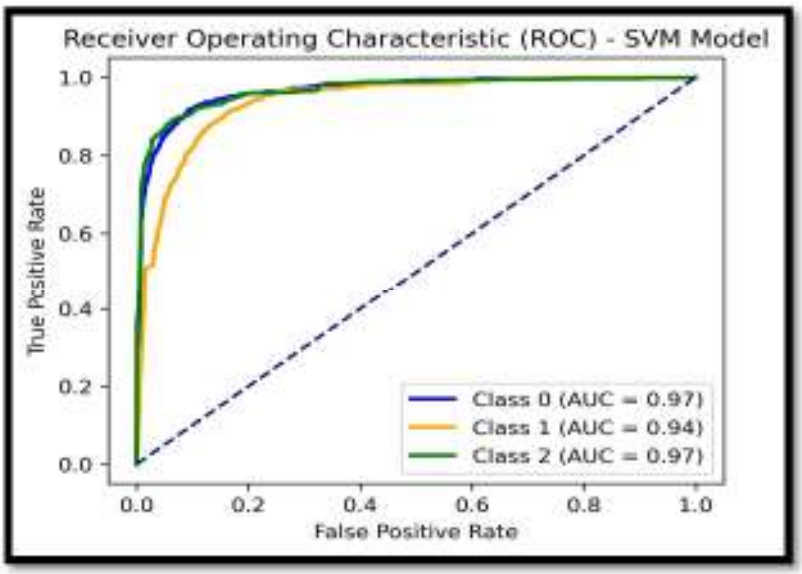
Class	AUC
Positive (2)	≃0.97
Neutral (1)	≃0.94
Negative (0)	≃ 0.97

The PR curves for the three classes are represented in Figure 3.3. These curves are informative when dealing with ambiguous or imbalanced classes, as they focus on the SVM classifier's ability to accurately identify pertinent occurrences.

Maintain a good precision for both positive and negative classes (stable performance), despite elevated recall levels, while Neutral exhibits more drop-off and demonstrates increased susceptibility to errors. And that highlights that while the SVM model performs well overall, the trade-off between precision and recall is delicate for the neutral class.

Figure 3.2

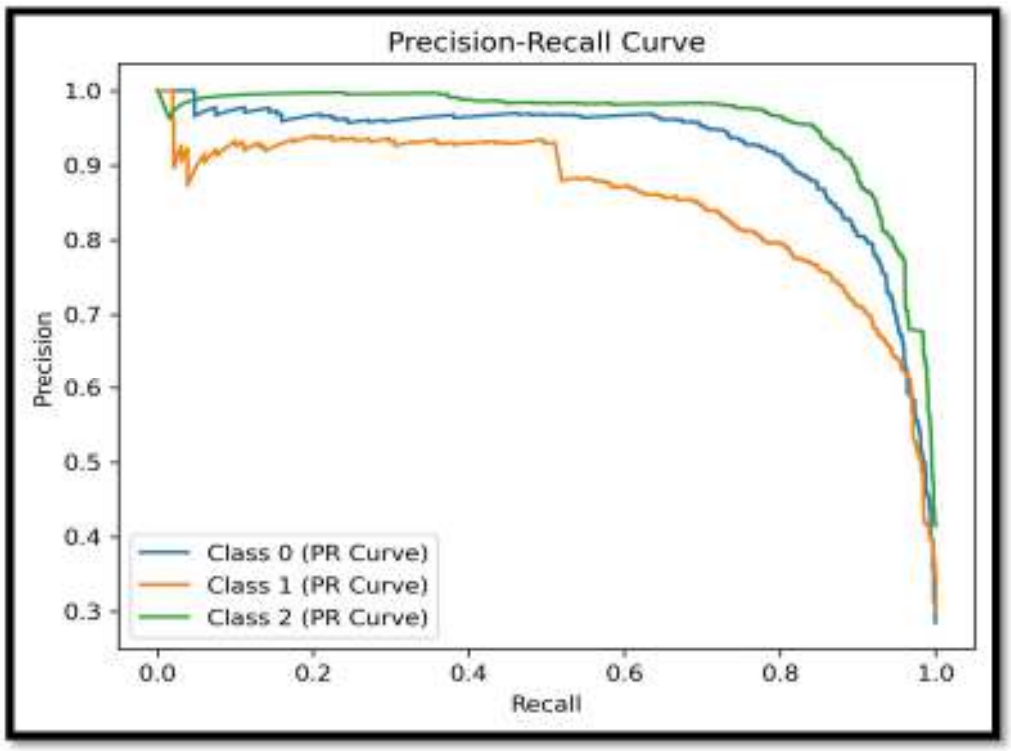
ROC curves for all sentiment classes. The SVM model exhibited robust class separability, achieving AUCs exceeding 0.94 across the board



The learning curve for both training and validation accuracy as a function of the size of the training set. The SVM classifier's training accuracy typically stays above 90%, and the validation accuracy perpetually ascends to approximately 85%, stabilizing. A modest yet consistent gap or disparity between training and validation suggests that the SVM model generalizes well without overfitting, and its performance further could benefit from additional training data.

Figure 3.3

Sentiment classes' PR curves



3.2 Recommendation System Results

3.2.1 Active users

ALS was used to test the sentiment-integrated model (RS+SA) and the baseline model (RS–SA) on active users to evaluate the effectiveness of sentiment-integrated recommendations. The evaluation was based on several performance metrics: RMSE, Precision@K, Recall@K, and NDCG@K, where K refers to the number of top-ranked items considered when computing the evaluation metric. The metrics were evaluated at multiple cutoff points: K = 5, 10, 20, and 30, as shown in the table below.

Table 3.3

Metric Comparison between RS–SA and RS+SA for active users

Metric	RS-SA				RS+SA			
	k=5	k=10	k=20	k=30	k=5	k=10	k=20	k=30
RMSE	0.2489	0.2453	0.2416	0.2416	0.0775	0.0773	0.0742	0.0746
Precision@K	0.2314	0.1250	0.0620	0.0414	0.2372	0.1273	0.0628	0.0415
Recall@K	0.6152	0.6322	0.6366	0.6366	0.6211	0.6335	0.6356	0.6403
NDCG@K	0.9367	0.9230	0.8804	0.8530	0.9611	0.9463	0.9201	0.9201

The results illustrate that the RS+SA model outperforms the RS–SA model consistently for both Precision@K and Recall@K across all K values. Although the modest improvements —for example, Precision@10 improves from 0.1250 to 0.1273, and Recall@10 improves from 0.6322 to 0.6335 —they show a steady upward trend in top-N recommendation performance when sentiment signals are included.

Notably, the improvements in Precision@K are more witnessed at lower K values, where high-ranking precision is most critical, since users typically look at and engage with only the top few recommended items. This means that the sentiment integration helps the model's ability to better rank and prioritize more personalized and relevant content at the top of the list, thereby enhancing the overall utility of the recommendation list.

While the differences in improvements in Recall@K are numerically tiny, they remain consistently in favor of RS+SA. This shows that the sentiment model is better at retrieving a wider range of relevant courses, enhancing coverage of the user's actual interests without compromising precision.

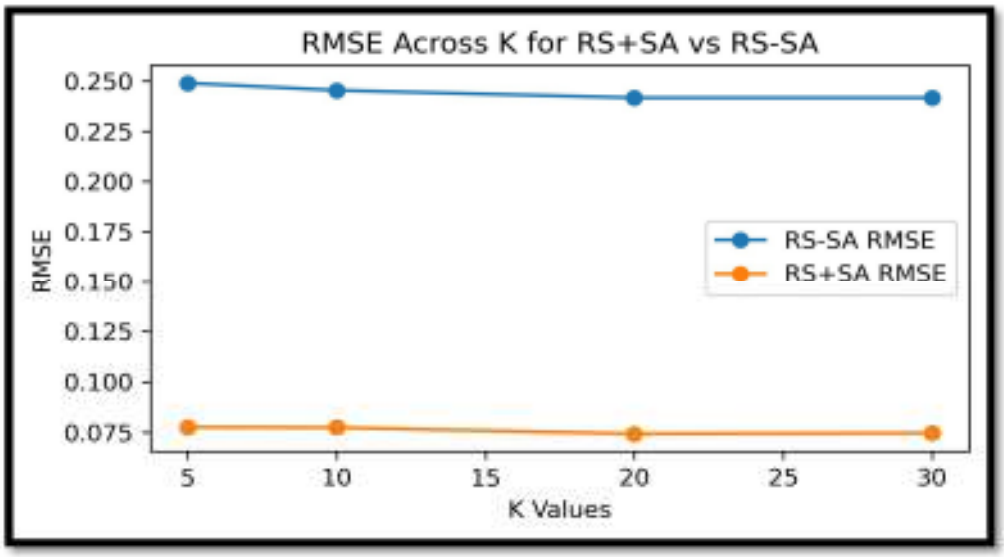
RS+SA shows higher NDCG scores among all K values, indicating its exceptional capacity to rank courses in more advantageous positions in addition to retrieving relevant courses. NDCG@K rewards placing relevant courses higher in the recommendation list. For example, in the case of NDCG@5, RS+SA achieves 0.9611 compared to 0.9367 for RS–SA, while for NDCG@30, RS+SA achieves 0.9201 compared to 0.8530 for RS–SA.

These findings suggest that the model can comprehend subtle user preferences, which leads to more context-aware ranking by including sentiment into user-item interactions. And this is important in e-learning platforms, where presenting the most suitable content at the top has a big impact on course enrollment and increases user satisfaction.

The RS+SA model illustrates a far lower RMSE compared to RS–SA even as K increases, as shown in Figure 3.4. The values are notably hovering around 0.074, while RS-SA keeps error levels close to 0.245. And that leads to a more generalizable and stable model, resistant to overfitting and more adept at learning from the enriched interaction matrix compared to RS-SA, which has a higher residual error. There is a nearly 70% decrease in RMSE, which suggests that this gap presents evidence that including sentiment improves rating predictions' accuracy and a better fit to user preferences.

Figure 3.4

RMSE metric comparing RS+SA with RS-SA across different K values



3.2.2 Cold users

To address the cold-users issue, a hybrid approach that integrates clustering and kNN techniques with sentiment-aware features. The results of this approach are summarized in the table below.

Table 3.4

Metric Comparison between RS–SA and RS+SA for cold users

Metric	RS-SA				RS+SA			
	k=5	k=10	k=20	k=30	k=5	k=10	k=20	k=30
Success Rate	0.0560	0.0668	0.1957	0.1583	0.0903	0.0927	0.9327	0.9331
Coverage	0.0915	0.1268	0.2254	0.6954	0.1514	0.2447	9.8310	15.8081
Diversity	0.9539	0.9877	0.9659	0.9725	0.9102	0.9474	0.9297	0.9456
Precision@K	0.0112	0.0067	0.0099	0.0053	0.0181	0.0093	0.0471	0.0315
Recall@K	0.0536	0.0636	0.1868	0.1506	0.0859	0.0871	0.9124	0.9141

The values for Precision@K show a noticeable improvement over all K values for the RS+SA model compared to the RS-SA model. For example, when Precision@5, RS+SA achieves 0.0181 compared to RS-SA with 0.0112. Also, when Precision@10, 0.0093 vs. 0.0067; when Precision@20, 0.0471 vs. 0.0099; and when Precision@30, 0.0315 vs. 0.0053. This pattern means that RS+SA performs better at recommending relevant items from the top-N list. In cold scenarios this is vital; the user engagement is affected by early ranking accuracy.

And the similar things to recall values, which show remarkable progress, when Recall@20, RS+SA achieves 0.9124 vs. 0.1868 with RS-SA, a nearly five-fold rise; when Recall@10, 0.0871 vs. 0.0636; and when Recall@5, 0.0859 vs. 0.0536. This increase in recall values indicates that RS+SA can better cover user preferences by returning a greater number of relevant recommendations.

Figure 3.5 illustrates the results obtained by the coverage metric. The RS-SA model shows at lower K values a very limited coverage. For example, when it K=5, only 0.0915 and when it K=10, only 0.1268. On the other hand, the RS+SA model reaching 0.1514 and 0.2447, respectively, at the same K values means it greatly expands the selection of recommended items. At higher K, the disparity becomes considerably more noticeable. For example, when it K=20, RS+SA achieves 9.8310 vs. 0.2254 for RS-SA and when it K=30, 15.8081 vs. 0.6954. This increase in coverage ensures better personalization and novelty, which is vital in cold scenarios where users have limited or no historical interactions, leading to sentiment integration, which leads to the increase of the system's exploratory capability in addition to improving accuracy.

Figure 3.5

Coverage metric comparing RS+SA with RS-SA across different K values

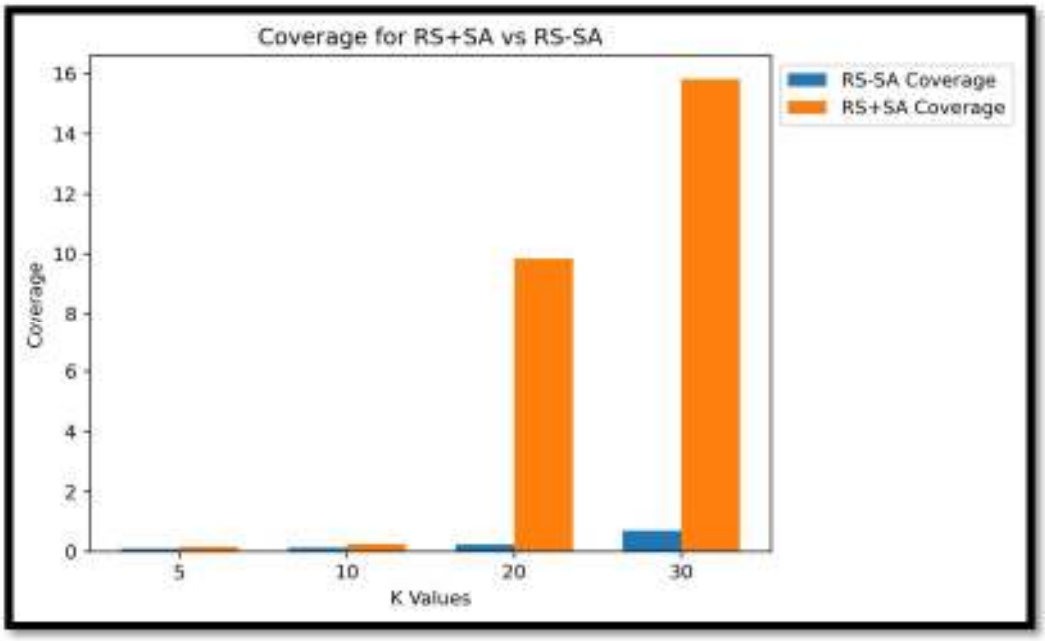
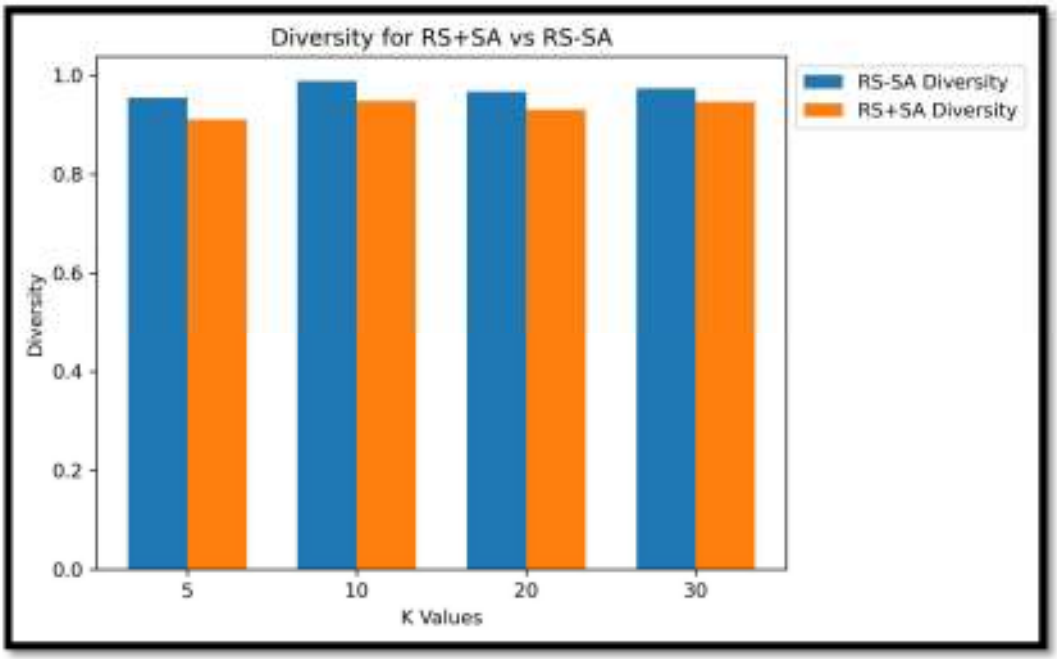


Figure 3.6 illustrates the results obtained by the diversity metric. The RS-SA model, across all K values, continuously obtains slightly higher diversity than RS+SA. For example, when k=10, RS-SA achieves 0.9877 compared to 0.9474 for RS+SA, when k=20, achieves 0.9659 vs. 0.9297, and when K=30, achieves 0.9725 vs. 0.9456. The RS+SA model tends more towards aligned sentiment recommendations, putting individuality and relevance ahead of maximum dispersion, as seen in this slight decrease in diversity. This decrease is offset by gains from coverage. The RS+SA highlights a meaningful trade-off between diversity and coverage. It's able to recommend more distinct item sets and courses to cold users while only slightly reducing diversity. This balance illustrates that recommendation of sentiments leads to more significant and refined recommendations without falling into repetitiveness, even if the diversity space is a little bit constrained (enough diversity).

Figure 3.6

Diversity metrics comparing RS+SA with RS-SA across different K values

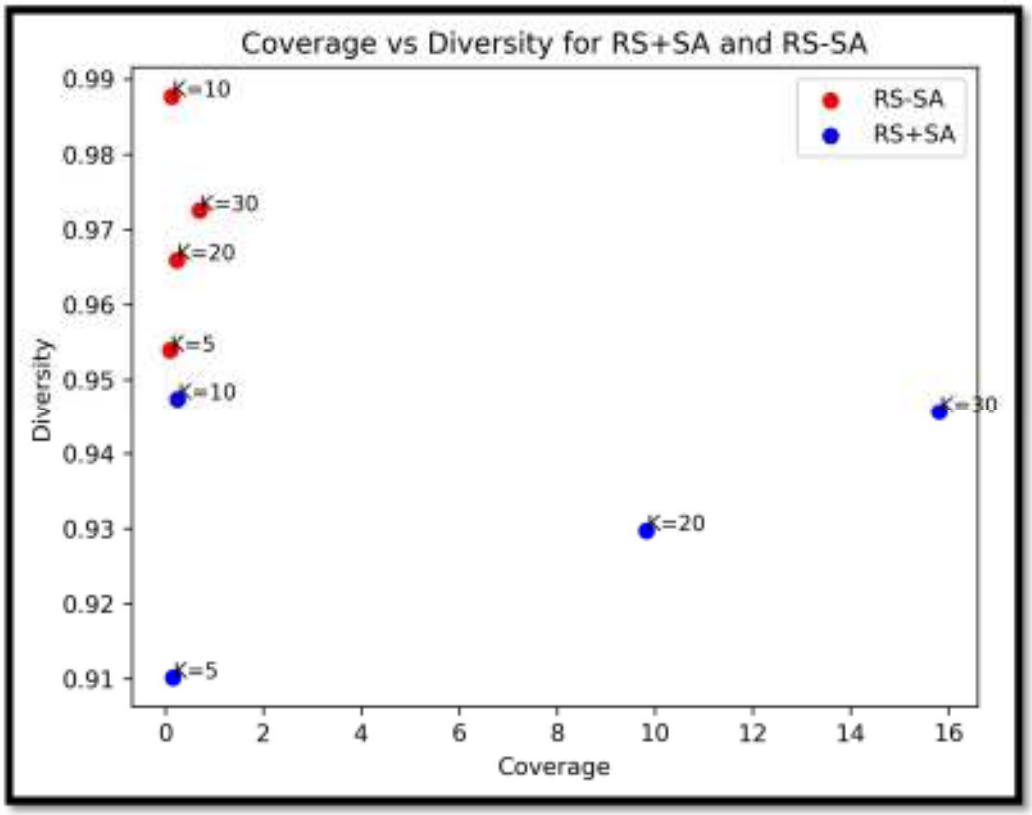


And as noticed, the RS-SA with red dots, particularly at higher K values, achieves higher diversity regularly, in return getting little value of coverage. In contrast, the RS+SA with blue dots, particularly when K increases, reaching over 15, notably increases coverage while preserving, across all K values, an acceptable level of diversity nearly above 0.91. And this demonstrates the trade-off between coverage and diversity for both models (RS-SA, RS+SA).

And this directs attention towards what sentiment integration makes. By exposing the users to a larger and more diverse pool of items while maintaining sufficient originality across recommendations. These coverage improvements offset the slight decline in diversity, which emphasizes that RS+SA is a more balanced and realistic model, especially for cold-start scenarios. This suggests that sentiment-enriched user modeling improves personalization and expands the reach and generalizability of the recommendation system.

Figure 3.7

Coverage vs. diversity metrics across different K values



The RS+SA model achieves a remarkable improvement across all K values in the success rate metric over the RS-SA model. For example, the RS+SA achieves a success rate above 93% when K = 20 or 30. On the other hand, the RS-SA achieves success rates below 20% and falls far behind. These improvements ensure that RS+SA is more able to provide more relevant and pleasant recommendations, even in cold start user scenarios where user profiles are limited.

The RS+SA model performs noticeably better than RS-SA across all sentiment classes. Users with positive and neutral sentiment show the notable gains compared to RS-SA. And this means that recommendations with sentiment, regardless of sentiment polarity, improve engagement for users with little to no previous interactions and more effectively personalize outcomes.

For the increase in cold users' success rate across all sentiment classes: positive, neutral, and negative. The big noticeable improvement was with positive sentiment, showing a 9.57% improvement, followed by neutral sentiment with 7.27% and negative sentiment with 7.87%. Even with sparse data, RS+SA effectively captures user preferences, as

evidenced by the consistent improvement observed across all sentiment classes. Its strength in cold-start scenarios is demonstrated by the fact that it performs excellently with positive users while simultaneously increasing personalization for neutral and negative ones.

3.3 Computational Cost Evaluation

To evaluate the computational cost of the proposed system across its major components, both training and prediction times were measured. The SA model using SVM displayed the highest training cost, requiring nearly 19 minutes and 31 seconds for prediction. Conversely, for active users the ALS-based RS was highly efficient, with only one second for training and 5 seconds for generating top-N recommendations. For cold users, clustering was completed in just 0.01 seconds, while generating recommendations for all cold users took 8 minutes. While sentiment classification and cold-user recommendation components incurred higher runtimes because of the complexity of features and hybrid method computations, the system as a whole remains practical for real-world use, where it can still be applied efficiently in e-learning platforms without the need of instant results.

Chapter Four

Conclusion and Future work

This study focuses on improving the Arabic e-learning experience by the integration of SA with CF RS. By making use of ML algorithms, particularly the SVM classifier, it was relied upon for sentiment classification for Arabic reviews and extraction of emotional input from students and incorporated it into the recommendation pipeline.

A comparison has been made between two models: a sentiment-aware model, RS+SA, and a baseline model without sentiment awareness, RS-SA. Experiments have shown that the RS+SA showed steady improvements in all important evaluation metrics, especially for cold-start users. These findings confirm that integrating sentiment closes the gap created by sparse interactions while also enhancing personalization.

For active students, RS+SA showed its capacity to fine-tune course recommendations through sentiment signals, achieving consistent improvements over the RS-SA baseline. In particular, RMSE was notably reduced by nearly 70%, and ranking metrics like NDCG@K improved from 0.9367 to 0.9611 at K=5, reflecting better recommendation accuracy and personalization. For cold start students, RS+SA performed across all K values noticeably better than the baseline model. Notable improvements are seen in the used metrics, such as success rate, coverage, and recall, particularly where at K=20, Recall@K increased from 0.1868 to 0.9124, success rate jumped from 0.1957 to 0.9327, coverage improved from 0.2254 to 9.8310, and diversity remained high from 0.9659 to 0.9297, proving the model not only improves accuracy but also provides varied and broad recommendations. These results strongly validate the integration of SA into RSs as a means to enhance both student modeling and cold-start issues.

Despite what has been achieved, the system will be able to get a lot of improvements in the future work. For example, a more detailed picture of learner preferences can be obtained by deeper integration of sentiment, such as aspect-based sentiment analysis. Also, the system's ability to expand to deal with non-Arabic content in addition to Arabic. Furthermore, expanding the study to a bigger dataset that contains, for example, the course duration or click patterns to combine temporal dynamics with user involvement signals to get more context-aware and adaptive recommendations. Future work could also look into DL-based recommendation models, which may further benefit from sentiment signals, including neural CF or graph-based RS.

List of Abbreviations

Abbreviation	Meaning
AI	Artificial Intelligence
ALS	Alternating Least Squares
API	Application Programming Interface
AUC	Area Under the Curve
AraBert	Arabic Bidirectional Encoder Representations from Transformers
ARM	Association Rule Mining
ASTD	Arabic Sentiment Tweets Dataset
AWS	Amazon Web Services
BERT	Bidirectional Encoder Representations from Transformers
BOW	Bag of Words
CB	Content-Based
CBOW	Continuous Bag of Words
CF	Collaborative Filtering
CHI	Chi-Square
CNN	Convolutional Neural Network
CSV	Comma-Separated Values
DL	Deep Learning
DT	Decision Tree
FPR	False Positive Rate
GA	Genetic Algorithm
GB	Gradient Boosting
GloVe	Global Vectors for Word Representation
HMM	Hidden Markov Model
IG	Information Gain
KNN	K-Nearest Neighbor
LDA	Latent Dirichlet Allocation
LFM	Latent Factor Model
LSTM	Long Short-Term Memory
LSVC	Linear Support Vector Classification
LR	Logistic Regression
MAE	Mean Absolute Error

MAP	Mean Average Precision
MDA	Multi-Criteria Decision Assistance
MI	Mutual Information
MLE	Maximum Likelihood Estimation
ML	Machine Learning
MLP	Multi-Layer Perceptron
MOOCs	Massive Open Online Courses
MNB	Multinomial Naive Bayes
MRR	Mean Reciprocal Rank
NMAE	Normalized Mean Absolute Error
NB	Naive Bayes
NDCG	Normalized Discounted Cumulative Gain
NLTK	Natural Language Toolkit
NLP	Natural Language Processing
NMF	Non-negative Matrix Factorization
NPTEL	National Program on Technology Enhanced Learning
NRC	National Research Council Lexicon
OOV	Out-of-Vocabulary
PCC	Pearson Correlation Coefficients
PR	Precision-Recall Curve
PSAU	Prince Sattam bin Abdulaziz University
RBF	Radial Basis Function
RF	Random Forest
ResNet	Residual Network
RNTN	Recursive Neural Tensor Network
ROC	Receiver Operating Characteristic
RMSE	Root Mean Square Error
RS	Recommendation System
S3	Simple Storage Service
S3VM	Semi-Supervised Support Vector Machine
SA	Sentiment Analysis
S-G	Skip Gram
SABCNN	Sentiment Analysis-Based Convolutional Neural Network

SL	Supervised Learning
SPADE	Sequential Pattern Discovery using Equivalence Classes
SPM	Sequential Pattern Mining
SSL	Semi-Supervised Learning
SVC	Support Vector Classification
SVD	Singular Value Decomposition
SVM	Support Vector Machine
SWAYAM	Study Webs of Active Learning for Young Aspiring Minds
TFR	Tri-Factor Recommendation
TF-IDF	Term Frequency-Inverse Document Frequency
TPR	True Positive Rate
UL	Unsupervised Learning
URL	Uniform Resource Locator
VADER	Valence Aware Dictionary and sEntiment Reasoner

References

[1] Li, S. (2023). The historical evolution of educational technology: From the printing press to online education. In *Proceedings of the 4th International Conference on Education Studies: Experience and Innovation (ICESEI 2023)* (Vol. 10). *Innovation Humanities and Social Sciences Research*.

[2] Morris, N. P., Ivancheva, M., Coop, T., Mogliacci, R., & Swinnerton, B. (2020). Negotiating growth of online education in higher education. *International Journal of Educational Technology in Higher Education*, *17*, 48.

[3] Salama, R., & Hinton, T. (2023). Online higher education: Current landscape and future trends. *Journal of Further and Higher Education*, *47*(7), 913-924.

[4] Dash, G., Akmal, S., Mehta, P., & Chakraborty, D. (2022). COVID-19 and e-learning adoption in higher education: A multi-group analysis and recommendation. *Sustainability*, *14*, 8799.

[5] Maatuk, A. M., Elberkawi, E. K., Aljawarneh, S., Rashaideh, H., & Alharbi, H. (2022). The COVID-19 pandemic and e-learning: Challenges and opportunities from the perspective of students and instructors. *Journal of Computing in Higher Education*, *34*, 21–38.

[6] Morshed, S. A., Khan, S. S., Tanvir, R. B., & Nur, S. (2021). Impact of COVID-19 pandemic on ride-hailing services based on large-scale Twitter data analysis. *Journal of Urban Management*, *10*, 155–165.

[7] Aini, Q., Budiarto, M., Putra, P. O. H., & Rahardja, U. (2020). Exploring e-learning challenges during the global COVID-19 pandemic: A review. *Jurnal Sistem Informasi*, *16*(2), 57–65.

[8] HR Vision. (n.d.). *Evolution of eLearning adoption*. HR Vision. https://www.hrvisionevent.com/content-hub/the-growing-importance-of-learning-management-systems/evolution-of-elearning-adoption/.

[9] Dhananjaya, G. M., Goudar, R. H., Kulkarni, A. A., Rathod, V. N., & Hukkeri, G. S. (2024). A digital recommendation system for personalized learning to enhance online education: A review. *IEEE Access*, *12*.

[10] Poornima, D., & Karthika, D. (2024). E-learning recommendation system and classification techniques—A survey. *Naturalista Campano*, *28*(1).

[11] Maryam, O., Ashraf, H., Amjad, T., & Jhanjhi, N. Z. (2023). Improved e-learning-based recommender systems: A survey [Preprint]. *Preprints*.

[12] Clarizia, F., Colace, F., De Santo, M., Lombardi, M., Pascale, F., & Pietrosanto, A. (2018). E-learning and sentiment analysis: A case study. In *Proceedings of the 6th International Conference on Information and Education Technology (ICIET '18)* (pp. 111–118).

[13] Rahman, M. A., Begum, M., Mahmud, T., Hossain, M. S., & Andersson, K. (2023). Analyzing sentiments in e-learning: A comparative study of Bangla and Romanized Bangla text using transformers. *IEEE Access*, *11*.

[14] Ezaldeen, H., Misra, R., Bisoy, S. K., Alatrash, R., & Priyadarshini, R. (2022). A hybrid e-learning recommendation integrating adaptive profiling and sentiment analysis. *Journal of Web Semantics*, *72*, 100700.

[15] Souabi, S., Retbi, A., Idrissi, M. K., & Bennani, S. (2021). Recommendation systems on e-learning and social learning: A systematic review. *The Electronic Journal of e-Learning*, *19*(5), 432–451.

[16] Shi, Z., & Agrawal, R. (2025). A comprehensive survey of contemporary Arabic sentiment analysis: Methods, challenges, and future directions [Preprint]. *arXiv*.

[17] Matrane, Y., Benabbou, F., & Sael, N. (2023). A systematic literature review of Arabic dialect sentiment analysis. *Journal of King Saud University – Computer and Information Sciences*, *35*(6), 101570.

[18] Karabila, I., Darraz, N., El-Ansari, A., Alami, N., & El Mallahi, M. (2023). Enhancing collaborative filtering-based recommender system using sentiment analysis. *Future Internet*, *15*, 235.

[19] Alduailej, A., & Alothaim, A. (2022). AraXLNet: Pre-trained language model for sentiment analysis of Arabic. *Journal of Big Data*, *9*, 72.

[20] Muhammad, I. (n.d.). Course Reviews on Coursera [Data set]. Kaggle. https://www.kaggle.com/datasets/imuhammad/course-reviews-on-coursera.

[21] Louati, A., Louati, H., Kariri, E., Alaskar, F., & Alotaibi, A. (2023). Sentiment analysis of Arabic course reviews of a Saudi university using support vector machine. *Applied Sciences*, *13*, 12539.

[22] Abo, M. E. M., Idris, N., Mahmud, R., Qazi, A., Hashem, I. A. T., Maitama, J. Z., Naseem, U., Khan, S. K., & Yang, S. (2021). A multi-criteria approach for Arabic dialect sentiment analysis for online reviews: Exploiting optimal machine learning algorithm selection. *Sustainability*, *13*, 10018.

[23] Essam, M., Elmenshawy, M., & Mousa, H. M. (2020). Arabic tweets sentiment analysis using hybrid approaches. *International Journal of Computer Applications*, *175*(36).

[24] Bessou, S., & Aberkane, R. (2019). Subjective sentiment analysis for Arabic newswire comments. *Journal of Digital Information Management*, *17*(5), 289–295.

[25] Heikal, M., Torki, M., & El-Makky, N. (2018). Sentiment analysis of Arabic tweets using deep learning. In *Proceedings of the 4th International Conference on Arabic Computational Linguistics (ACLing 2018), Dubai, United Arab Emirates* (pp. 114–122). *Procedia Computer Science*, *142*, 114–122.

[26] Rajesh, P., & Suseendran, G. (2020). Prediction of N-gram language models using sentiment analysis on e-learning reviews. In *2020 International Conference on Intelligent Engineering and Management (ICIEM)*. IEEE.

[27] Kechaou, Z., Ben Ammar, M., & Alimi, A. M. (2011). Improving e-learning with sentiment analysis of users' opinions. In *2011 IEEE Global Engineering Education Conference (EDUCON) – Learning Environments and Ecosystems in Engineering Education*. IEEE.

[28] AL-Rubaiee, H., Qiu, R., Alomar, K., & Li, D. (2016). Sentiment analysis of Arabic tweets in e-learning. *Journal of Computer Science*, *12*(11), 553–563.

[29] Ali, M. M. (2021). Arabic sentiment analysis about online learning to mitigate COVID-19. *Journal of Intelligent Systems*, *30*, 524–540.

[30] Obeidat, R., Duwairi, R., & Al-Aiad, A. (2019). A collaborative recommendation system for online courses recommendations. In *2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML), Istanbul, Turkey* (pp. 49–54).

[31] Rani, L. P. J., Wise, D. C. J. W., Ajayram, K. V., Gokul, T., & Kirubakaran, B. (2020). Course recommendation for students using machine learning. In *Proceedings of the International Conference on Electronics and Sustainable Communication Systems (ICESC 2020)*. IEEE.

[32] Han, Y., & Zhong, W. (2024). Personalized recommendation of English online teaching content based on a logistic regression algorithm. *Journal of Electrical Systems*, *20*(6s), 1925–1936.

[33] Joshi, N., & Gupta, R. (2020). A personalized web-based e-learning recommendation system to enhance user learning experience. *International Journal of Recent Technology and Engineering (IJRTE)*, *9*(1), 1186.

[34] Subha, S., Sankaralingam, B. P., Gurusamy, A., Sehar, S., & Bavirisetti, D. P. (2023). Personalization-based deep hybrid e-learning model for online course recommendation system. *PeerJ Computer Science*, *9*, e1670.

[35] Talaghzi, J., Bellafkih, M., Bennane, A., Himmi, M. M., & Amraouy, M. (2023). A combined e-learning course recommender system. *International Journal of Emerging Technologies in Learning (iJET)*, *18*(6).

[36] Gotardo, R. A., Hruschka Junior, E. R., Zorzo, S. D., & Cereda, P. R. M. (2013). Approach to cold-start problem in recommender systems in the context of web-based education. In *2013 12th International Conference on Machine Learning and Applications (ICMLA)*. IEEE.

[37] Butmeh, H., & Abu-Issa, A. (2024). Hybrid attribute-based recommender system for personalized e-learning with emphasis on cold start problem. *Frontiers in Computer Science, 6*, 1404391.

[38] El Maazouzi, Q., Retbi, A., & Bennani, S. (2024). Enhancing online learning: Sentiment analysis and collaborative filtering from Twitter social network for personalized recommendations. *International Journal of Electrical and Computer Engineering (IJECE)*, *14*(3), 3266–3276.

[39] Al-Ajlan, A., & Alshareef, N. (2023). Recommender system for Arabic content using sentiment analysis of user reviews. *Electronics*, *12*, 2785.

[40] Dang, C. N., Moreno-García, M. N., & Prieta, F. D. L. (2021). An approach to integrating sentiment analysis into recommender systems. *Sensors*, *21*, 5666.

[41] Zamri, N., Palanichamy, N., & Haw, S.-C. (2023). College course recommender system based on sentiment analysis. *Advanced Science, Engineering and Information Technology*, *13*(5).

[42] Ziani, A., Azizi, N., Schwab, D., Aldwairi, M., & Chekkai, N. (2017). Recommender system through sentiment analysis. In *Proceedings of the 2nd International Conference on Automatic Control, Telecommunications and Signals, Annaba, Algeria*.

[43] Mawane, J., Naji, A., & Ramdani, M. (2020). Recommender e-learning platform using sentiment analysis aggregation. In *Proceedings of the 2020 International Conference on Intelligent Systems and Advanced Computing (SITA'20)*, Rabat, Morocco. ACM.

[44] Joundy Hazar, M., Zrigui, M., & Maraoui, M. (2022). Learner comments-based recommendation system. In *26th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2022)*. *Procedia Computer Science*, *207*, 2000–2012.

[45] Alatrash, R., Ezaldeen, H., Misra, R., & Priyadarshini, R. (n.d.). Sentiment analysis using deep learning for recommendation in e-learning domain. In *Springer Book Chapter*.

[46] Ezaldeen, H., Misra, R., Bisoy, S. K., Alatrash, R., & Priyadarshini, R. (2021). A hybrid e-learning recommendation integrating adaptive profiling and sentiment analysis. *Web Semantics: Science, Services and Agents on the World Wide Web*, *72*, 100700.

[47] Hazar, M. J., Jaballi, S., Maraoui, M., Zrigui, M., & Nicolas, H. (2025). A hybrid e-learning recommendation system incorporating user reviews and ratings for enhanced course selection [Preprint]. *Research Square*.

[48] nlptown. (2023). *bert-base-multilingual-uncased-sentiment* [Pretrained sentiment model]. Hugging Face. https://doi.org/10.57967/hf/1515.

[49] Alammary, A. S. (2022). BERT models for Arabic text classification: a systematic review. *Applied Sciences*, *12*(11), 5720.

[50] Abdelgwad, M. M., Soliman, T. H. A., & Taloba, A. I. (2022). Arabic aspect sentiment polarity classification using BERT. *Journal of Big Data*, *9*(1), 115.

[51] aubmindlab. (n.d.). *aubmindlab/bert-base-arabertv02* [Pretrained model]. Hugging Face. https://huggingface.co/aubmindlab/bert-base-arabertv02.

[52] Hegazi, M. O., Al-Dossari, Y., Al-Yahy, A., Al-Sumari, A., & Hilal, A. (2021). Preprocessing Arabic text on social media. *Heliyon*, *7*(2).

[53] Almutairi, T., Saifuddin, S., Alotaibi, R., Sarhan, S., & Nassif, S. (2024). Preprocessing Techniques for Clustering Arabic Text: Challenges and Future Directions. *International Journal of Advanced Computer Science & Applications*, *15*(8).

[54] Nassr, Z., Sael, N., & Benabbou, F. (2020). Preprocessing arabic dialect for sentiment mining: State of art. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *44*, 323-330.

[55] Obeid, O., Zalmout, N., Khalifa, S., Taji, D., Oudah, M., Alhafni, B.,... & Habash, N. (2020, May). CAMeL tools: An open source python toolkit for Arabic natural language processing. In *Proceedings of the twelfth language resources and evaluation conference* (pp. 7022-7032).

[56] El-Khair, I. A. (2017). Effects of stop words elimination for Arabic information retrieval: a comparative study. *arXiv preprint arXiv:1702.01925*.

[57] Abuhammad, A. S. (2024). Negation Detection In Arabic Opinion Reviews: A Comprehensive Annotated Dataset For Sentiment Analysis. *Journal of Information Systems Research and Practice*, 2(4), 2–19.

[58] Alyafeai, Z., Al-shaibani, M. S., Ghaleb, M., & Ahmad, I. (2021). Evaluating Various Tokenizers for Arabic Text Classification. arXiv preprint arXiv:2106.07540.

[59] Taghva, K., Elkoury, R., & Coombs, J. (2005). Arabic Stemming without a Root Dictionary. Information Science Research Institute, University of Nevada, Las Vegas, USA.

[60] El-Shishtawy, T., & El-Ghannam, F. (2012). An Accurate Arabic Root-Based Lemmatizer for Information Retrieval Purposes. *arXiv preprint arXiv:1203.3584*.

[61] Mubarak, H. (2017). Build Fast and Accurate Lemmatization for Arabic. *arXiv preprint arXiv:1710.06700*.

[62] Das, M., & Alphonse, P. J. A. (2023). A comparative study on tf-idf feature weighting method and its analysis using unstructured dataset. *arXiv preprint arXiv:2308.04037*.

[63] Jalilifard, A., Caridá, V. F., Mansano, A. F., Cristo, R. S., & da Fonseca, F. P. C. (2021). Semantic sensitive TF-IDF to determine word relevance in documents. In *Advances in Computing and Network Communications: Proceedings of CoCoNet 2020, Volume 2* (pp. 327-337). Springer Singapore.

[64] Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, *5*, 135-146.

[65] Almuayqil, S. N., Humayun, M., Jhanjhi, N. Z., Almufareh, M. F., & Javed, D. (2022). Framework for improved sentiment analysis via random minority oversampling for user tweet review classification. *Electronics*, *11*(19), 3058.

[66] Sayyed, Z. A. (2021). Study of sampling methods in sentiment analysis of imbalanced data. *arXiv preprint arXiv:2106.06673*.

[67] Obi, J. C. (2023). A comparative study of several classification metrics and their performances on data. *World Journal of Advanced Engineering Technology and Sciences*, *8*(1), 308-314.

[68] Davis, J., & Goadrich, M. (2006, June). The relationship between Precision-Recall and ROC curves. In *Proceedings of the 23rd international conference on Machine learning* (pp. 233-240).

[69] Viering, T., & Loog, M. (2022). The shape of learning curves: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*(6), 7799-7819.

[70] Sathyanarayanan, S., & Tantri, B. R. (2024). Confusion matrix-based performance evaluation metrics. *African Journal of Biomedical Research*, 4023-4031.

[71] Wikipedia contributors. (2025, March 14). *Confusion matrix*. Wikipedia. https://en.wikipedia.org/wiki/Confusion_matrix.

[72] Hu, Y., Koren, Y., & Volinsky, C. (2008, December). Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE international conference on data mining* (pp. 263-272). Ieee.

[73] Adyatma, H. A., & Baizal, Z. K. A. (2023). Book recommender system using matrix factorization with alternating least square method. *Journal of Information System Research (JOSH)*, *4*(4), 1286-1292.

[74] Kulkarni, A. (n.d.). *Matrix factorization with alternating least squares*. Retrieved May 17, 2025, from https://akshay-a-kulkarni.github.io/reports/ALS.pdf

[75] AL-Bakri, N. F., & Hashim, S. H. (2019). Collaborative filtering recommendation model based on k-means clustering. *Al-Nahrain Journal of Science*, *22*(1), 74-79.

[76] Mohammed, S. A. (2020). Adaptation of k-means clustering algorithm for collaborative filtering based recommendations. *Tilburg University school of Humanities and Digital Sciences Department of Cognitive Science & Artificial Intelligence Tilburg, The Netherlands*.

[77] Anwar, T., Uma, V., Hussain, M. I., & Pantula, M. (2022). Collaborative filtering and kNN-based recommendation to overcome cold start and sparsity issues: A comparative analysis. *Multimedia tools and applications*, *81*(25), 35693-35711.

[78] Januzaj, Y., Beqiri, E., & Luma, A. (2023). Determining the Optimal Number of Clusters using Silhouette Score as a Data Mining Technique. *International Journal of Online & Biomedical Engineering*, *19*(4).

[79] Zhang, S., Yao, L., Sun, A., & Tay, Y. (2019). Deep learning based recommender system: A survey and new perspectives. *ACM computing surveys (CSUR)*, *52*(1), 1-38.

[80] Vargas, S., & Castells, P. (2011, October). Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the fifth ACM conference on Recommender systems* (pp. 109-116).

[81] Schein, A. I., Popescul, A., Ungar, L. H., & Pennock, D. M. (2002, August). Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 253-260).

[82] Bell, R. M., & Koren, Y. (2007, August). Improved neighborhood-based collaborative filtering. In *KDD cup and workshop at the 13th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 7-14). sn.

[83] Fernández-Tobías, I., Tomeo, P., Cantador, I., Di Noia, T., & Di Sciascio, E. (2016, September). Accuracy and diversity in cross-domain recommendations for cold-start users with positive-only feedback. In *Proceedings of the 10th ACM conference on Recommender Systems* (pp. 119-122).

[84] Panteli, A., & Boutsinas, B. (2023). Addressing the cold-start problem in recommender systems based on frequent patterns. *Algorithms*, *16*(4), 182.

[85] Yuan, H., & Hernandez, A. A. (2023). User cold start problem in recommendation systems: A systematic review. *IEEE access*, *11*, 136958-136977.

[86] Al-Twairesh, N., Al-Khalifa, H., Alsalman, A., & Al-Ohali, Y. (2018). Sentiment analysis of arabic tweets: Feature engineering and a hybrid approach. *arXiv preprint arXiv:1805.08533*.

[87] El-Beltagy, S. R., & Ali, A. (2013, March). Open issues in the sentiment analysis of Arabic social media: A case study. In *2013 9th International Conference on Innovations in information technology (IIT)* (pp. 215-220). IEEE.

Appendices

Appendix A

Summary of Related Research on Hybrid Recommendation Systems

Ref	Dataset	Lang.	Feature Extraction	Sentiment Analysis	Recommendation System	Evaluation
[38]	Twitter elearning reviews, Online Course Data	English	TF-IDF, BoW	NB, SVM, RF	KNN, PCC	Accuracy = 95% (SVM), RMSE = 0.683
[39]	LABR	Arabic	-	Mazajak, Arabic BERT-Mini, AraBERT	Memory-based CF (Cosine, Euclidean, Manhattan) Model-based CF (SVD, KNN, NMF)	RMSE = 0.580, MAE = 0.42 (SVD, Arabic BERT-Mini)
[40]	Amazon Food Reviews, Amazon Movie Reviews	English	BERT embedding	hybrid DL (CNN, LSTM)	SVD, NMF, SVD++	Rating Prediction (Best β = 0.3, Movie Reviews, SVD++) MAE: 0.5770, RMSE: 0.8577, NMAE= 0.1443 Top-N Recommendation (Best β = 0.7, Food Reviews, SVD++) MRR = 84.33%, MAP = 73.84%, NDCG = 86.89%
[41]	College Course, Student Feedback Dataset	English	N-grams, TF-IDF	Lexicon-based (VADER, TextBlob, Flair)	KNN, Fuzzy Logic	Accuracy=85.71% (Fuzzy Logic, TF-IDF, N-gram)

[42]	English dataset, French Dataset, Arabic/Algerian dataset	Arabic, Algerian, French, English	lexical and syntactic features	S3VM	user-based CF (Spearman similarity)	MAE: 0.50-0.60, Precision: 90% - 100%, Recall: 100%
[43]	Coursera MOOC datasets	English	TF-IDF	Lexicon-based (TF-IDF), Rule-based Filtering	Hybrid (Unsupervised DL, CF)	-
[44]	Coursera	English	LDA, TF-IDF, LFM	VADER	Hybrid Model (CBF, DL CNN)	Cosine Similarity 0.98
[45]	Book reviews scraped from Amazon with corresponding ratings	English	GloVe	CNN	SABCNN-based recommendation model	accuracy = 77%
[46]	ABHR	English	S-G, CBOW	CNN	Hybrid (CF, CB)	accuracy=89.1% (SA)
[47]	Coursera	English	LDA, LFM	CNN	Hybrid (CF, CB)	98% similarity between original and predicted rating

Appendix B

Core Algorithms Pseudocode Used in the System

B.1 *SA SVM Pseudo Code*

Algorithm 1 SA SVM Pseudo Code

Load and preprocess text reviews (e.g., cleaning, normalization)
Extract TF-IDF and FastText features
Concatenate features into a unified representation
Define sentiment labels (*Positive, Neutral, Negative*) using BERT
Initialize the Support Vector Machine (SVM) classifier
for each fold in 5-fold cross-validation **do**
    Split the data into training and validation sets
    Train the SVM classifier on the training set
    Evaluate the model on the validation set and store metrics
end for
Compute average performance across all folds
Predict sentiment class for each review
Assign numerical sentiment scores based on predicted class
Output sentiment class and sentiment score for each review

B.2 *CF ALS-based Pseudo Code*

Algorithm 2 CF ALS-based Pseudo Code

Load preprocessed training data containing user-item interactions
Construct a sparse user-item interaction matrix
Initialize ALS (Alternating Least Squares) parameters:
    1. Number of latent factors
    2. Regularization term
    3. Number of iterations
Train the ALS model on the weighted user-item matrix
for each active user **do**
    Generate top-N course recommendations using the trained ALS
    Filter out already interacted items
end for
Evaluate model performance using metrics:
    1. RMSE
    2. Precision@K
    3. Recall@K
    4. NDCG@K
Output ranked course recommendations for each active user

B.3 *Cold User Recommendation using K-Means and Hybrid KNN Strategy Pseudo Code*

Algorithm 3 K-Means and Hybrid KNN Strategy Pseudo Code

Load user profiles and their sentiment-enhanced feature vectors
Apply K-Means clustering to group users based on feature similarity
Assign each cold user to the nearest cluster based on their profile
for each cold user **do**
    Identify top-K most similar users from the assigned cluster using KNN
    Recommend top-K courses based on interactions from those similar users
    Extract all courses within the cluster and vectorize them using TF-IDF
    Apply item-based KNN using cosine similarity between the user's preferred course and courses in the cluster
    Recommend top-K most similar courses based on content similarity
end for
Combine recommendations from user-based and item-based KNN
Filter out any previously seen items for the cold user
Rank recommendations by relevance
Evaluate recommendation quality using:

1. Precision@K
2. Recall@K
3. Success Rate
4. Coverage
5. Diversity

Output top-N personalized course recommendations for each cold user

Appendix C

Tables

Table C.1

Coursera Courses File

Variable	Class	Description
name	character	The name of the course
institution	character	The designation of the educational institution or platform associated with the course under examination
course_url	character	Course URL
course_id	character	Course ID

Table C.2

Coursera Reviews File

Variable	Class	Description
reviews	character	Course review
reviewers	character	The identity of the reviewer who authored the review
date_reviews	date	Date of review publication
rating	integer	The evaluation score assigned by the reviewer to the course (1-5)
course_id	character	Course ID

Table C.3

Class Distribution prior to upsampling

Sentiment Class	Label	Count
Positive	2	8600
Neutral	1	5370
Negative	0	6026

Table C.4

Interaction Score Feature Components

Feature	Description
Normalized Rating	Rating supplied by the student, scaled from 0 to 1
Normalized Sentiment Score	Sentiment derived from the student's review content, normalized on a scale from 0 to 1
Sentiment Trend	The trajectory of sentiment variation over time for each user
Recency Decay	Exponential degradation contingent upon the age of the review
Review Length	Word count in the review
User-Course Ratio	Proportion of student engagement to course popularity
Average Course Rating	Average rating of the course among all students

Table C.5

ALS hyperparameters with tuned values

Hyperparameter	Description	Value
factors	Quantity of latent factors for user/item vectors	150
regularization	L2 regularization to mitigate overfitting	1
iterations	Count of ALS optimization iterations	10

Table C.6

Manually Preserved Arabic Sentiment Phrases for Tokenization

Original Phrase	Preserved form with Underscore	Sentiment
غير مقبول	غير_مقبول	Negative
جيد جدا	جيد_جدا	Positive
غير ضروري	غير_ضروري	Negative
سيء للغاية	سيء_للغاية	Negative
غير واضح	غير_واضح	Negative
أنصح به	أنصح_به	Positive

Table C.7

Hybrid Selective Stemming and Lemmatization

Token	In Lemmatization_words?	Operation	Output
الكورس	No	Stemming	كورس
الدورات	Yes	Lemmatization	الدورات
التعليمية	Yes	Lemmatization	التعليمية
أنصح	No	Stemming	نصح

Appendix D

Figures

Figure D.1

Sentiment distribution obtained from user-submitted star ratings. Ratings were categorized into sentiment classifications based on established thresholds (4–5 stars as positive, 3 stars as neutral, and 1–2 stars as negative)

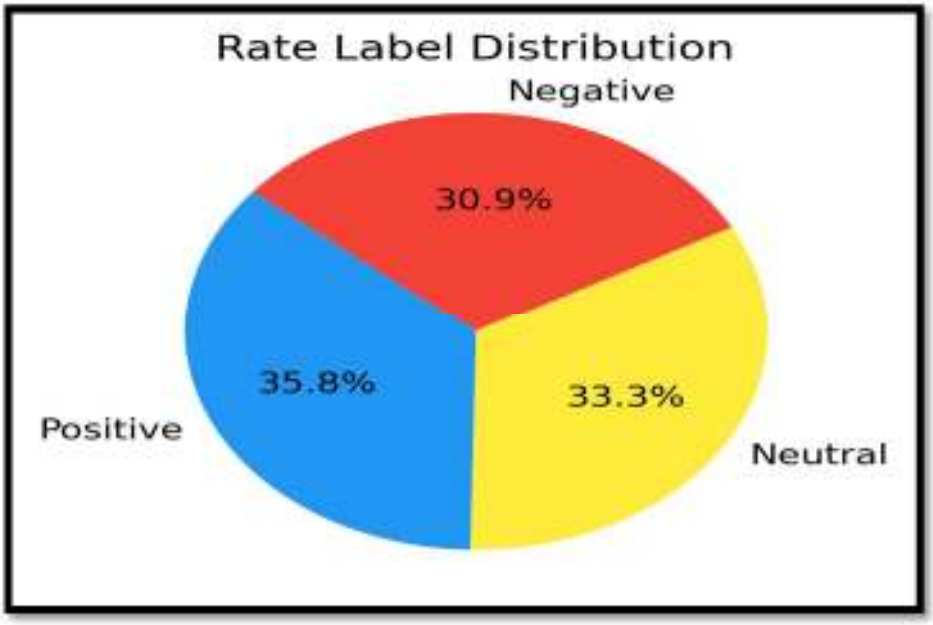


Figure D.2

Distribution of sentiment classifications produced by the fine-tuned BERT model. The algorithm categorizes Arabic course reviews into positive, neutral, and negative classifications based on the textual content

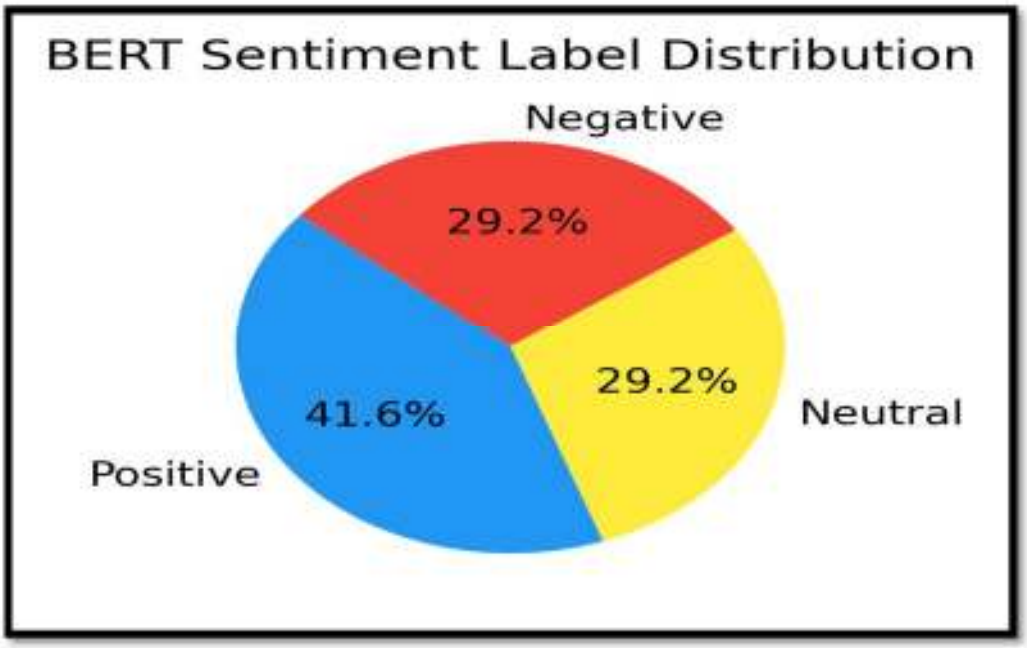


Figure D.3

Distribution of sentiment classes before applying the upsampling for the purpose of balancing the training dataset

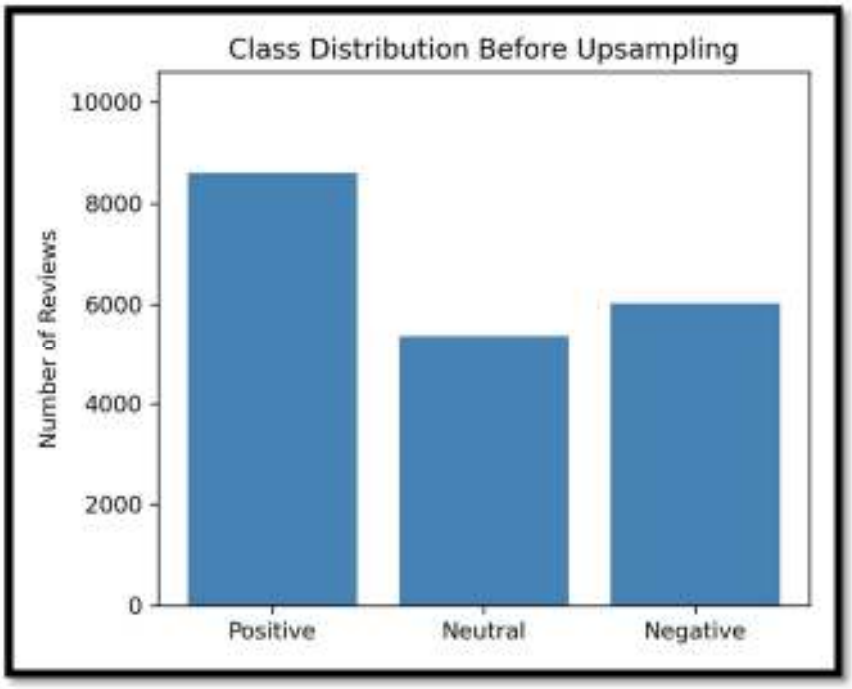


Figure D.4

Distribution of sentiment classes after applying the upsampling for the purpose of balancing the training dataset

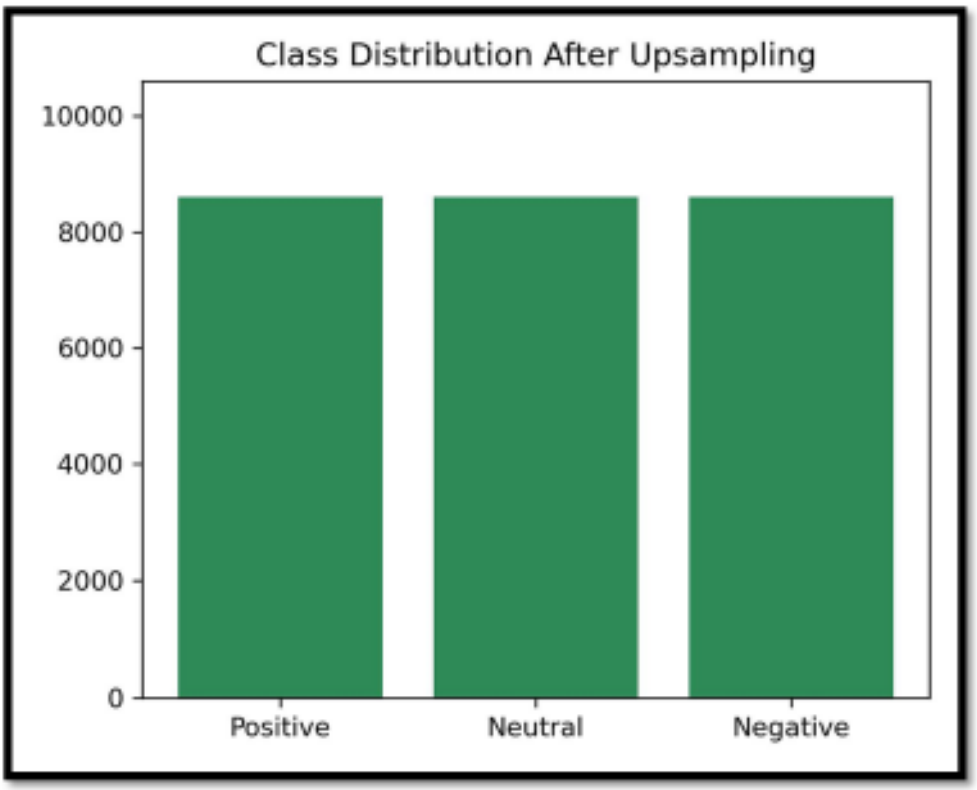
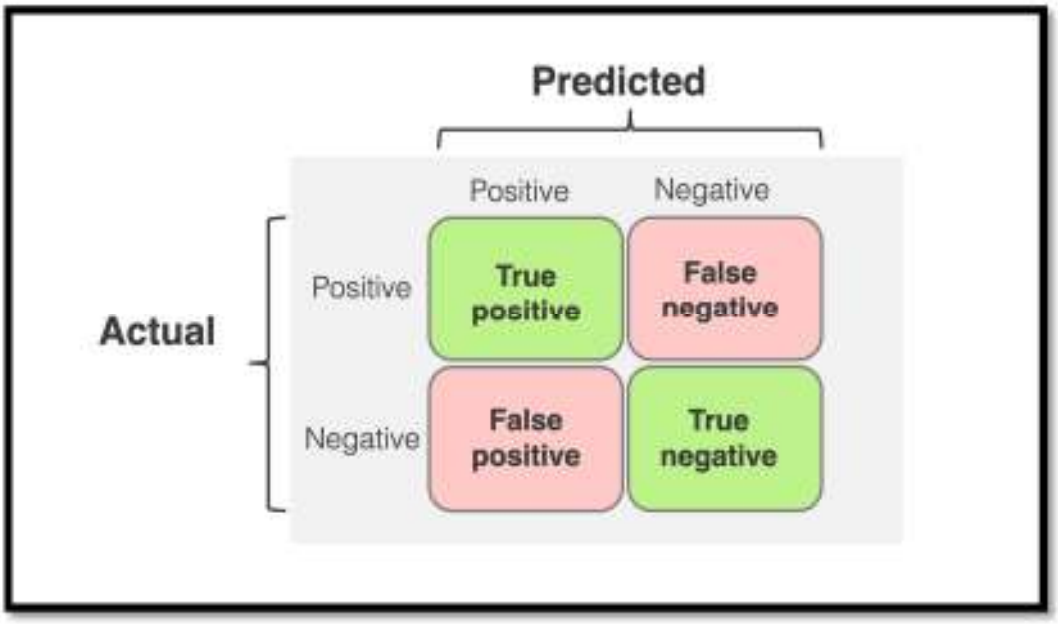


Figure D.5

Confusion Matrix Configuration for Binary Classification: This figure shows the correlation between actual and expected classes. It classifies predictions into four categories: true positive (TP), false negative (FN), false positive (FP), and true negative (TN). The rows refer to the actual labels, and the columns refer to the expected labels. Green cells refer to the accurate predictions, while the red cells refer to the significant errors



Note: Adopted from [71]

Figure D.6

Silhouette score for different cluster amounts (K = 2 to 10). The peak score was observed at K=2, indicating the most distinct clustering configuration for cold-start user profiling

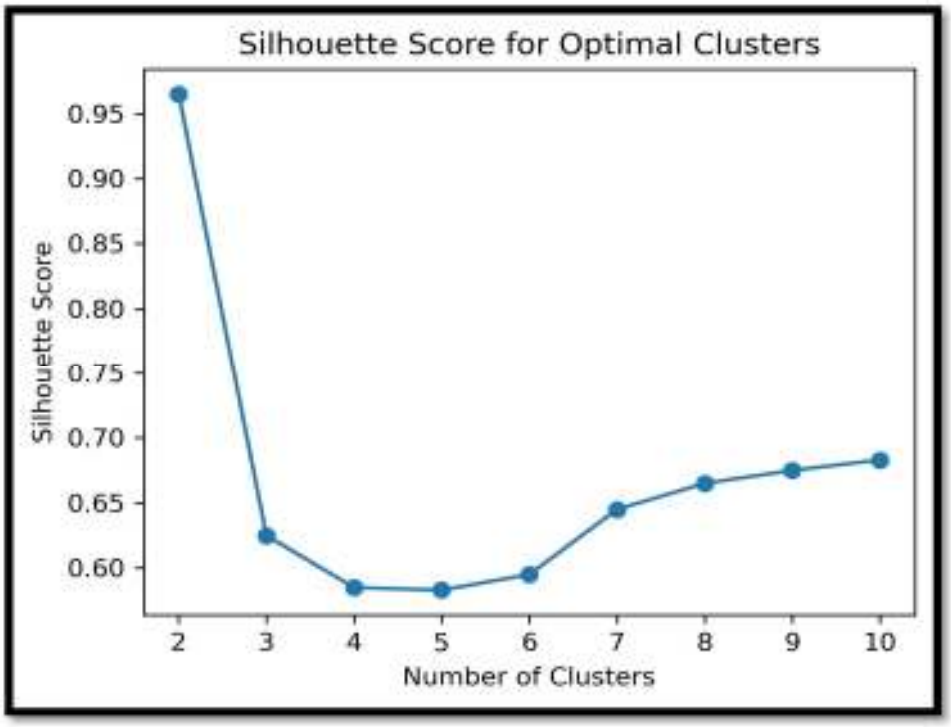


Figure D.7

Normalized Confusion Matrix

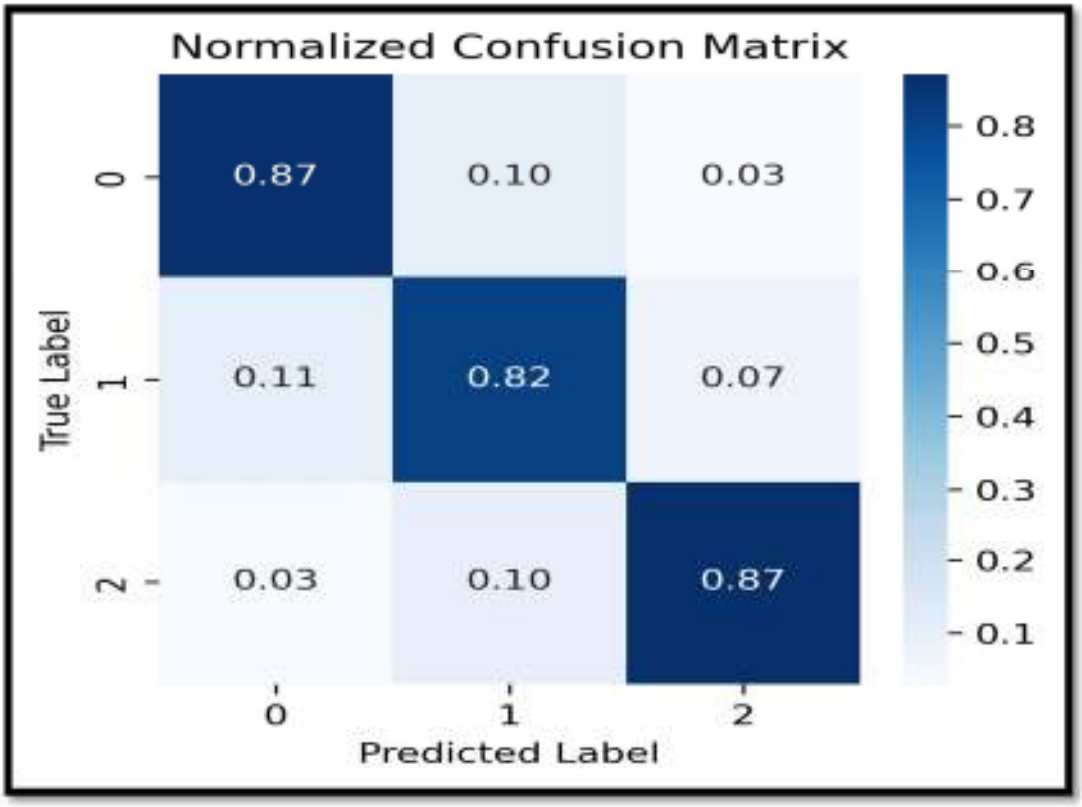


Figure D.8

Learning curve training and validation accuracy as sample numbers increase

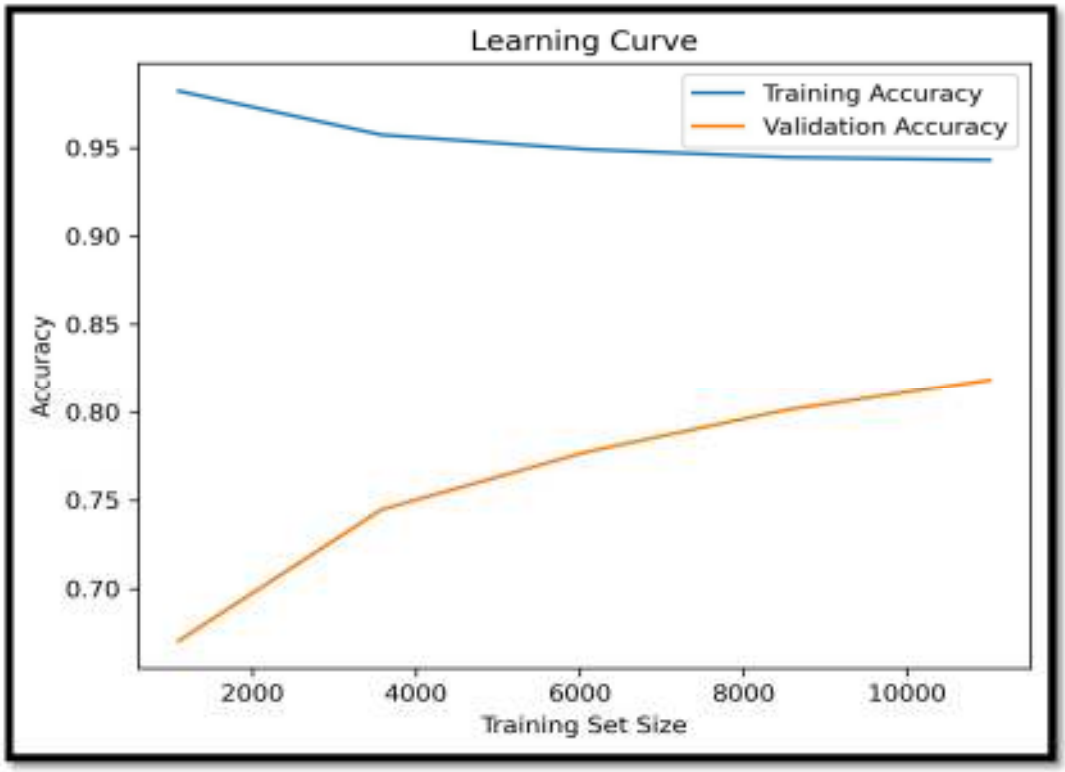


Figure D.9

Precision metric comparing RS+SA with RS-SA across different K values

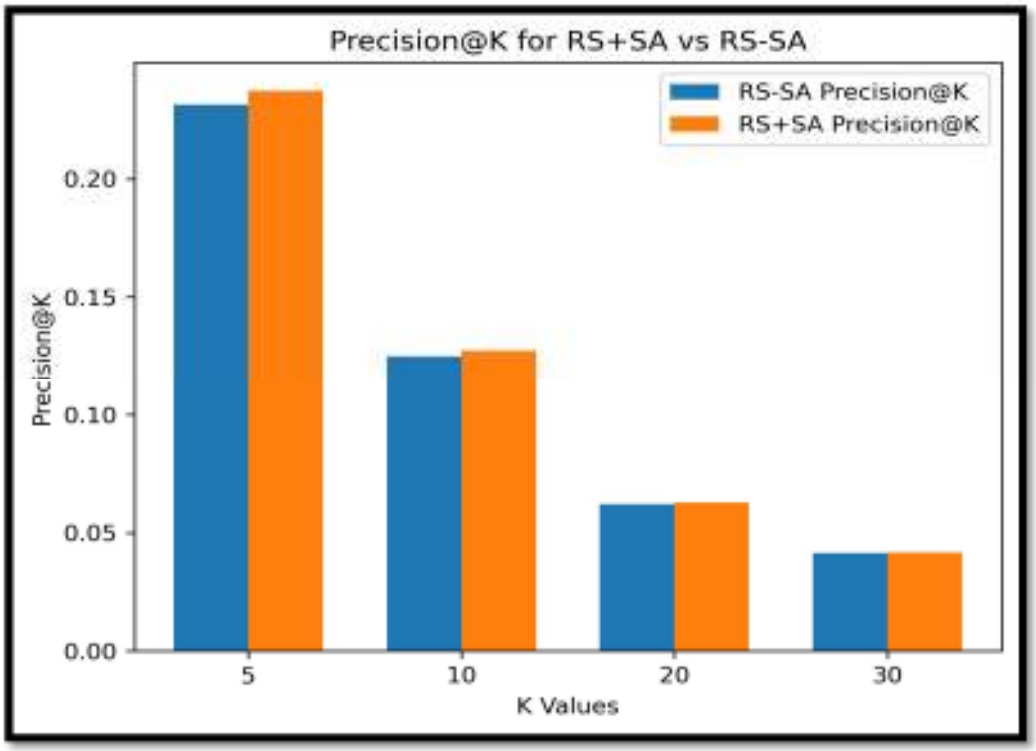


Figure D.10

Recall the metric comparing RS+SA with RS-SA across different K values

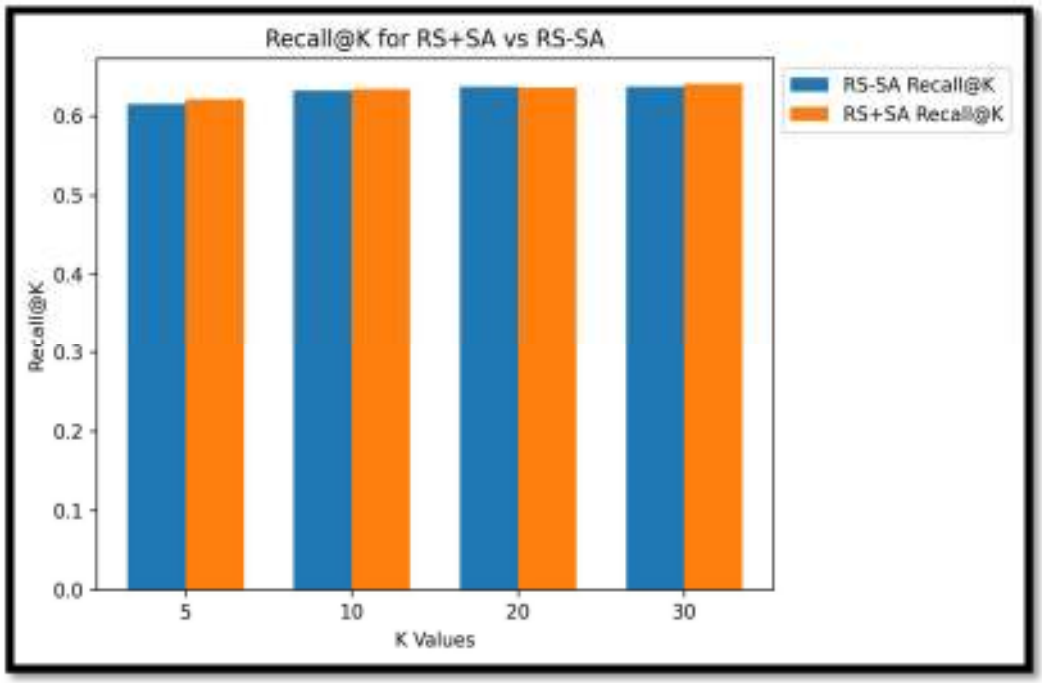


Figure D.11

NDCG metric comparing RS+SA with RS-SA across different K values

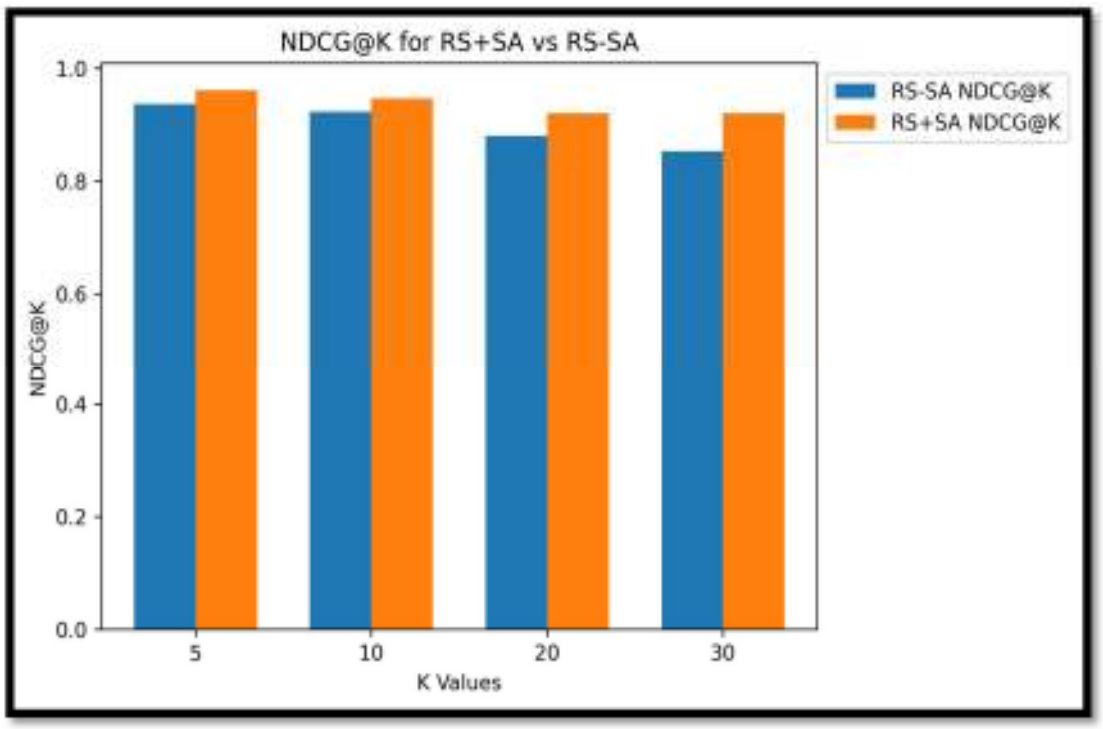


Figure D.12

Success rate metric comparing RS+SA with RS-SA across different K values

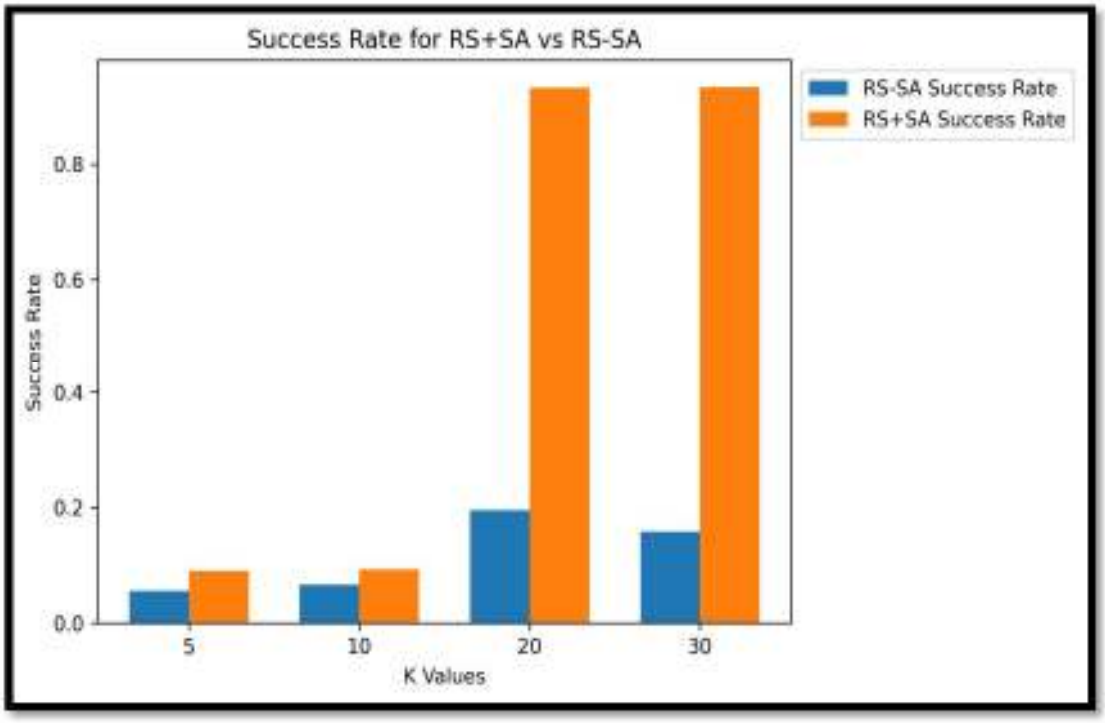


Figure D.13

Success Rate for Cold Users by Sentiment Contribution: RS+SA vs RS-SA

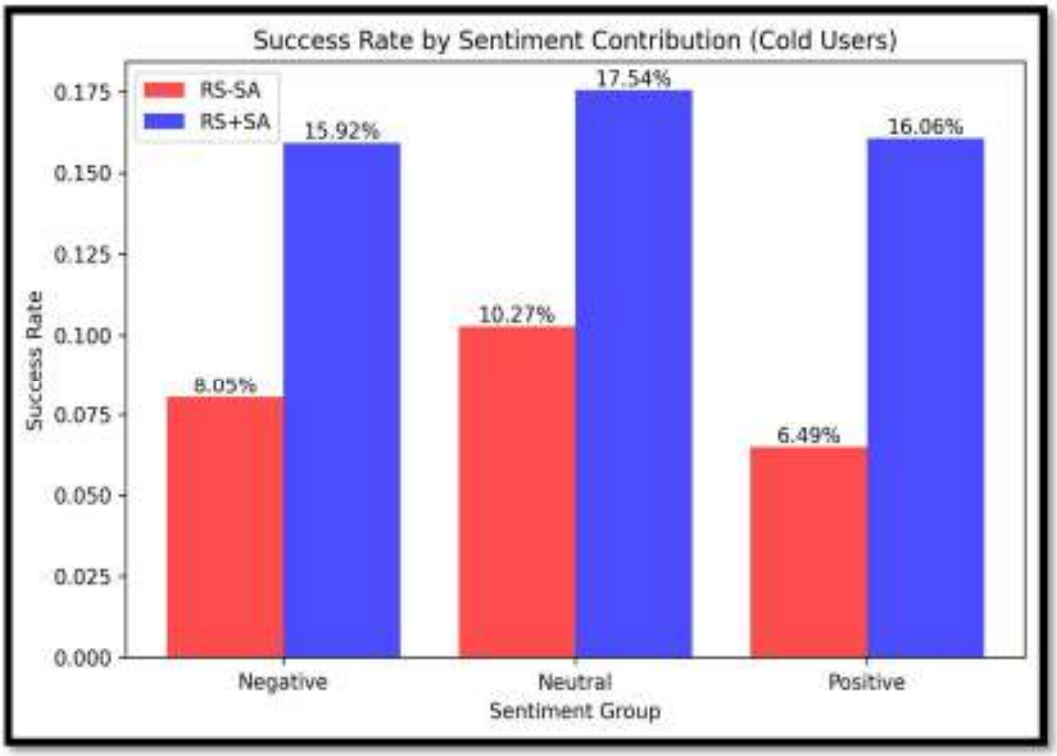
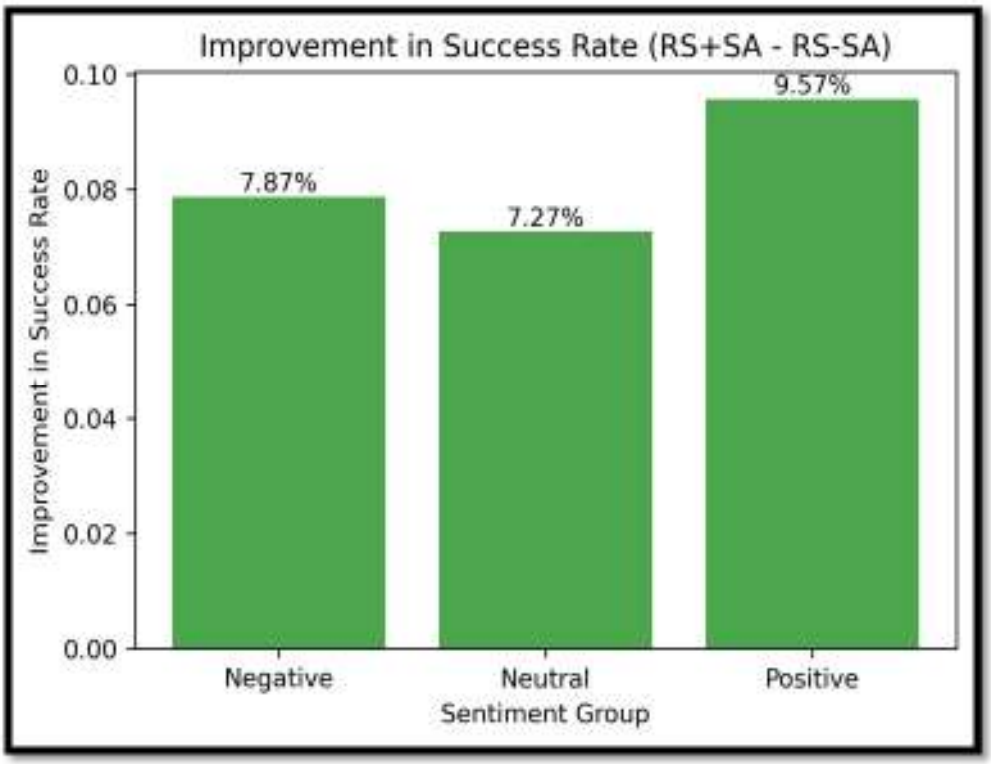


Figure D.14

Increase in Cold User Success Rate by Sentiment Group (RS+SA – RS-SA)



جامعة النجاح الوطنية

كلية الدراسات العليا

تحسين تجربة المستخدم في منصات التعليم الإلكتروني العربية باستخدام تحليل المشاعر وتقنيات التصفية التعاونية

إعداد

آية سعيد يامين

إشراف

د. عماد النتشة

قدمت هذه الرسالة استكمالا لمتطلبات الحصول على درجة الماجستير في الذكاء الاصطناعي،
من كلية الدراسات العليا، في جامعة النجاح الوطنية، نابلس – فلسطين.

2025

تحسين تجربة المستخدم في منصات التعليم الإلكتروني العربية باستخدام تحليل المشاعر وتقنيات التصفية التعاونية

إعداد

آية سعيد يامين

إشراف

د. عماد النتشة

الملخص

تهدف هذه الدراسة إلى تطوير نظام توصية محسّن (RS) بغرض تحسين تجربة المستخدمين في بيئات التعلم الإلكتروني، من خلال تحقيق التكامل بين تحليل المشاعر (SA) وتقنيات التصفية التعاونية (CF). وقد تم استخدام مجموعة بيانات عامة باللغة الإنجليزية من منصة Coursera، ومن ثم تم ترجمتها إلى اللغة العربية باستخدام خدمات الترجمة من AWS لتكون مناسبة لسياق اللغة المستهدفة. تستهدف الدراسة مراجعات الدورات التدريبية المقدّمة باللغة العربية، مع الأخذ بعين الاعتبار تحديات ندرة البيانات وتعقيد اللغة. تم استخدام نموذج محسن متعدد اللغات من نوع BERT (نماذج التمثيل ثنائي الاتجاه من المحولات) لتوليد تسميات المشاعر، في حين تم تطبيق خوارزمية آلة الدعم الناقل (SVM) بالاعتماد على مزيج من ميزات TF-IDF وFastText، وحققت أداءً جيدًا في تصنيف المشاعر. بعد ذلك، تم إثراء مصفوفة التفاعل بين المستخدم والعنصر بالنقاط المستخرجة من تحليل المشاعر، ليتم استخدامها في نظام التوصية القائم على خوارزمية التحليل التبادلي الأدنى (ALS) للمستخدمين النشطين، بينما تم التعامل مع المستخدمين الجدد (الباردين) من خلال تجميعهم باستخدام خوارزمية K-means بالاعتماد على الملفات الشخصية، يليها تطبيق نموذج هجيني يجمع بين أقرب الجيران (KNN) والتشابه باستخدام TF-IDF لأسماء الدورات التدريبية باللغة العربية. تم تقييم النظام من خلال إجراء مقارنة قبل وبعد دمج تأثير المشاعر بشكل منفصل لكل من المستخدمين النشطين والباردين. يستهدف النظام على وجه الخصوص المستخدمين أو المتعلمين الذين يخططون لدراسة

الدورات التدريبية باللغة العربية على المنصات الإلكترونية. وقد أظهرت النتائج أن هذا التكامل مع معلومات المشاعر يساهم في تقليل القيود المرتبطة بمشكلة المستخدم البارد، كما يعزز من مستوى التخصيص في التوصيات. فعلى سبيل المثال، حقق النموذج المعزز بالمشاعر انخفاضًا في قيمة RMSE بنسبة تقارب 70٪ للمستخدمين النشطين، وتحسناً ملحوظًا في معدل النجاح (من 19.57٪ إلى 93.27٪) والاستدعاء (من 18.68٪ إلى 91.24٪) للمستخدمين الباردين. وتؤكد هذه التحسينات على أن النظام المقترح يُعد حلاً عمليًا وفعّالًا لمنصات التعلم الإلكتروني.

الكلمات المفتاحية: تحليل المشاعر، التصفية التعاونية، نظام التوصية، التعلم الإلكتروني، النصوص العربية، مشكلة البداية الباردة، BERT متعدد اللغات، آلة الدعم الناقل (SVM).