International Journal for Housing Science and Its Applications
Publish August 3, 2025. Volume 46, Issue 3 Pages 676-691

https://doi.org/10.70517/ijhsa46350

Design of K-Nearest Neighbor Algorithm Based Music Course Content Recommendation System for Colleges and Universities Driven by Music Sentiment Analysis

Yaping Zeng[1,*]
[1] School of Music, Zhengzhou Preschool Education College, Zhengzhou, Henan, 450000, China
Corresponding authors: (e-mail: zengyaping@zzpec.edu.cn).

Abstract With the rapid development of the Internet, traditional teaching methods have been unable to meet the learning needs of college teachers and students. First, we use network crawler technology to collect research data, complete data preprocessing work through a series of operations such as word splitting, deactivation of words, feature extraction, etc., and input the processed data into polynomial plain Bayesian classifier for training to realize the classification and analysis of music emotion features. On this premise, with the help of similarity algorithm and K nearest neighbor algorithm, the music course content recommendation algorithm is constructed. With the support of this paper's algorithm and related development software, the design of music course content recommendation system in colleges and universities is completed, and the system is empirically analyzed. Compared with other systems, the real-time update delay and real-time recommendation delay of this paper's system are shorter, the update delay is less than 1000ms, and the corresponding recommendation delay is less than 500, which verifies that this paper's system has excellent operational performance, can bring students and teachers a comfortable experience, and promote the development of intelligent music teaching in colleges and universities.

Index Terms k-nearest neighbor algorithm, music recommendation system, similarity, plain Bayesian

I. Introduction

With the rapid development of information technology, the importance of digital teaching resources in college education has become increasingly prominent. Especially in the design of music curriculum, effective development and utilization of digital curriculum resources is the key to improve teaching quality and student participation [1]. In this context, the digital transformation of music curriculum in colleges and universities is not only a technological update, but also an innovation of the educational model and teaching concept. The digitization of modern curriculum resources can greatly enrich the content of music courses and provide a more intuitive and interactive way of learning, thus better attracting students' interest and participation [2], [3]. However, achieving this transformation requires educators to conduct an in-depth analysis of existing curriculum resources, identify their shortcomings, and explore how to establish a curriculum content recommendation system through technological means to effectively integrate and optimize these resources [4], [5].

At this stage, the music system on the market generally adopts the method of recommending for the user's interest in the category of collection music selected by the user when registering, which can satisfy a lot of user needs [6], [7]. However, when the user's mood changes, the music they want to listen to will also change partially, and then the music tracks recommended by this recommendation method will not be able to meet the user's needs for music [8], [9]. In response to the real-time changes in user mood, the method of sentiment analysis is added to the collection of user listening preferences, which can recommend the music that the user is interested in to the user's song list, which greatly improves the accuracy of the results of the recommended music, and saves a lot of valuable time for the user to find music [10]-[13].

Music, as an art reflecting the emotions of human real life, is an effective means of expressing emotions, as well as people's psychological feelings, and a symbol to convey their own joys and sorrows. More and more researchers take emotional attributes as a key part of music retrieval to build a more accurate music content recommendation system. Literature [14] proposes a music recommendation system eSM based on sentiment intensity metrics, which can fully consider the change of user's sentiment intensity by feature extraction of review text through a sentiment analysis framework, so as to make song recommendations. Literature [15] developed a new index for emotion-based music recommendation system, i.e., by establishing a “SWEMS” indexing mechanism to extract the “emotional patterns” of music, which helps to quantify the subjective emotional experience of the user and help the user search for music content that matches his/her emotions faster. This helps to quantify the subjective emotional experience of users and help them search for music content that matches their emotions faster. Literature [16] carries out sentiment analysis on users' social media comments to extract users' emotional polarity, and thus constructs a user sentiment prediction model, which is applied to the music recommendation system to effectively meet users' music sentiment needs. Literature [17] constructs personalized descriptive information such as the results of user emotion analysis as pairs of labels, and by establishing the adjacency between music and labels, it can satisfy the user's personalized demand for music. Literature [18] emphasizes that the user's emotion is also one of the important factors in their choice of music, through wearable physiological sensors can accurately collect the user's emotional state data, which can be input into the music recommendation engine as supplemental data, which can effectively improve the recommendation accuracy of the system. Literature [19] utilizes sensor technology to recognize facial expressions to detect the corresponding emotional state, and incorporates them into the filter to establish an emotion-based music content recommendation method, which achieves a high recommendation effect while avoiding the tedious operation of manual feature extraction. Literature [20] establishes a facial expression recognition model based on deep neural networks to integrate the user's emotion analysis results with music emotion data to generate a list of music recommendations that match the user's current emotions. In this context, the introduction of an intelligent system based on music sentiment analysis into the music course content recommendation system can improve the effectiveness of music education, which allows students to select learning content based on their own interests and emotions, and realizes personalized and flexible music teaching design.

In this paper, the initial data is collected using web crawler technology, and the basic preprocessing operations are carried out on this data, which include word splitting, deactivation of words, and feature extraction. The processed data is put into the plain Bayesian classifier for training, and problems such as unstable values and loss of accuracy appear. In view of the above problems, it is proposed to use Laplace smoothing and other techniques to make corrections, and finally complete the design of sentiment analysis algorithm based on user comments. Then a music course content recommendation model is constructed under the joint effect of K nearest neighbor algorithm and similarity algorithm, and in order to make the model better serve the music courses in colleges and universities, accordingly, a music course content recommendation model based on sentiment analysis algorithm and KNN algorithm is designed. Finally, an experimental simulation environment is constructed, relevant evaluation indexes are set, and the system in this paper is verified and analyzed, with a view to promoting the intelligent and digital innovation of music courses in colleges and universities.

II. Research on Recommendation Algorithms Driven by Music Emotion

II. A. Sentiment analysis algorithm based on user comments

Before proposing and introducing new attributes for hybrid recommendation, it is first necessary to get the new attributes, and this section describes in detail the process of getting the sentiment attributes of a song, which can be divided into three processes: review text acquisition, text preprocessing, and sentiment classification. The technical principles needed in the sentiment analysis process are also studied and introduced in depth.

II. A. 1) Text acquisition

Before performing sentiment analysis, data needs to be acquired, and in this paper, web crawlers are used to acquire the data. A web crawler is a software program that automatically searches, crawls, and extracts the content of a website, also known as a web spider, web robot, or web spider [21]. Web crawlers are divided into two main parts, crawler: responsible for downloading web content from a website and interacting with the web server via Hypertext Transfer Protocol (HTTP) or other network protocols. Parser: responsible for parsing the downloaded web content, extracting useful information and storing the information in a database or file.

(1) The workflow of web crawler

The crawler program initiates a request to the web server via HTTP to request web content. Then the web server returns the web content to the crawler program. Next, the crawler program parses the web page content and extracts useful information. The crawler stores the extracted information in a database or file. Finally the crawler program extracts a new Uniform Resource Locator (URL) from the web page and continues crawling, parsing and storing. While performing web crawling, it is necessary to follow the website's robot protocols, i.e., it is important to comply with the website's rules for the use of robots while crawling the website's content to avoid unnecessary load or damage to the website.

Web crawlers have a wide range of applications in the fields of natural language processing, data mining, web information searching, website monitoring, etc. However, there are some legal, ethical and privacy issues that need to be considered and solved when performing web crawling.

(2) Web crawling methods

Depth-first search: depth-first search is a graph-theoretic algorithm that will visit all points in the graph in the order of edges starting from the starting point until all reachable nodes are searched. In web crawlers, depth-first search is commonly used to search website content in a specific order.

(3) Main Web Crawler Libraries

Requests, Selenium and BeautifulSoup are third-party libraries for Python that are used to implement web crawlers. Use Requests to send HTTP requests for static data. Use Beautiful Soup to parse the document and extract the required data, process the crawled data, save it to a local file or database, and use multi-threaded or distributed architecture to improve the efficiency of the crawler.

II. A. 2) Text pre-processing

After crawling the text of user comments, then the text is preprocessed, the basic preprocessing operations are: word splitting, deactivation of words, feature extraction and other processes.

(1) Segmentation

Chinese word separation technology is divided into dictionary-based methods and statistics-based methods, and this paper uses statistics-based methods to complete the task of Chinese word separation. Statistical-based Chinese word segmentation technology is based on language modeling, which predicts the unknown words in the text through statistics on a large amount of corpus. The main process of this technique is: to establish a language model, using a large amount of corpus to count the information such as word frequency, correlation relationship between words and so on, so as to establish a language model. Utilize the language model for word segmentation, for the text to be segmented, word by word from left to right, and predict whether the generated words are correct or not by the language model. Calculate the likelihood of the participle result, for each generated word, calculate the probability of the word appearing using the language model. Select the most probable participle scheme and select the most probable words from left to right to get the final participle result.

(2) De-duplication

De-duplication is a common data preprocessing step in the field of natural language processing. The purpose of this process is to remove some words that do not have obvious semantic meaning from the original text, such as articles, pronouns, conjunctions and so on. The process of removing deactivated words usually starts with the creation of a deactivation lexicon, which is a corpus storing deactivated words that can be used to compare the words in the text. Then the deactivation word removal process is done by comparing the words in the deactivation dictionary after word splitting and removing the words that match the deactivation dictionary from the text. Finally the remaining words are organized into a new text and stored in the desired format.

(3) Feature Extraction

Feature extraction operations can effectively extract important feature words in the text and use these feature words for sentiment analysis. Both bag-of-words model and TF-IDF can be used to extract the feature words of the text, but their ideas and implementation methods are different. Bag-of-words model mainly focuses on the frequency of words in the text, while TF-IDF focuses more on the importance of words in the text. And TF-IDF is more suitable for sentiment analysis, so this paper adopts TF-IDF for feature extraction [22].

The formula of TF-IDF algorithm is:

$$TF-IDF(t,d)=TF(t,d)*IDF(t) \tag{1}$$

where t is the word, d is the document, $TF(t,d)$ is the frequency of occurrence of the word t in the document d, and $IDF(t)$ is the inverse document frequency of the occurrence of the word t in the set of documents. The word frequency (TF) can be used to measure the importance of words in a text and can be calculated using the following formula:

$$TF(t,d)=\frac{n_{t,d}}{\sum_{i\in d} n_{i,d}} \tag{2}$$

where the numerator represents the number of times the word t appears in the document d and the denominator is the length of the document d, i.e., the total number of words in the document d.

The inverse document frequency can be used to measure the uniqueness of a word in a collection of text and can be calculated using the following formula:

$$IDF(t)=log\frac{|N|}{|\{j:t\in d_j\}|} \tag{3}$$

where $|N|$ is the total number of documents in the document collection and the denominator represents the number of documents in which the word t occurs in the document collection. Ultimately, the weight value of each word, i.e., the TF-IDF value, is calculated by multiplying the word frequency with the inverse document frequency.

II. A. 3) Polynomial Plain Bayes-based Sentiment Classification

After the completion of the text preprocessing operations can be carried out in the text sentiment analysis process of sentiment classification, sentiment classification is the most important step in the process of sentiment analysis, the results of the classification of the direct impact of the system's strengths and weaknesses, so it is necessary to study the appropriate way to classify the sentiment. Plain Bayesian classification performs well on small datasets because it does not require too much data to support its assumptions. So for user comment text messages, it is more appropriate to use plain Bayesian classifier for sentiment classification. The Bayesian formula on which the plain Bayesian classifier is based is as follows:

$$P(C \mid X) = \frac{P(C)P(X \mid C)}{P(X)} \tag{4}$$

where C denotes the category, X denotes the feature, $P(C)$ denotes the prior probability, $P(X \mid C)$ denotes the posterior probability, and $P(X)$ denotes the probability normalization factor. According to the independence assumption of the plain Bayesian classifier, the posterior probability can be expressed as $P(X \mid C) = P(x1 \mid C)P(x2 \mid C) \cdots P(xn \mid C)$ where x_i is a dimension in the features, and $P(xi \mid C)$ denotes the probability of x_i under the category C.

Plain Bayesian classifiers produce numerical instability in the case of rare events, which is due to the loss of accuracy due to too small values of the probabilities during the computation process. To solve this problem, techniques such as Laplace smoothing are usually used to correct it. The plain Bayesian classifier is a simple and effective classification algorithm that has good prospects for application when the assumption of feature independence holds, but still needs further improvement in the case of multidimensional features. The formula theorem for the polynomial plain Bayesian classifier is as follows:

For a given text feature $x = x(x1, x2, \cdots, xn)$ and category y, the conditional probability distribution of the feature x can be approximated as a polynomial distribution given the category y, i.e.:

$$P(x \mid y) = \frac{\alpha + \sum_{i=1}^{n} x_i}{N + \alpha d} \tag{5}$$

where α is the smoothing parameter, $\sum_{i=1}^{n} x_i$ is the number of occurrences of x in the given category, N is the sum of the occurrences of all features in the given category y, and d is the total dimensionality of the features with $\alpha > 0$. The decision rule for the polynomial plain Bayesian classifier is the same as the decision rule for the plain Bayesian classifier, i.e.:

$$y = \arg\max y_i \{P(y_i)P(x \mid y_i)\} \tag{6}$$

where y_i is a category in the set of categories, $P(y_i)$ is the prior probability of a given category y_i, and $P(x \mid y_i)$ is the conditional probability of a given category y_i. Polynomial plain Bayesian classifiers take into account more fully the interactions between features by adding terms that are powers of the number of features, resulting in more accurate classification results. However, polynomial plain Bayesian classifiers are prone to overfitting when dealing with high-dimensional sparse data, so some smoothing techniques, such as Laplace smoothing, are needed to prevent overfitting. In practice, polynomial plain Bayesian classifier is a simple and efficient text classification method, which is widely used in the fields of sentiment analysis and text classification.

II. B.Music course content recommendation algorithm

After obtaining the sentiment categorization of the user reviews, the K-nearest neighbor algorithm is used to predict the ratings of the target users for the unrated albums. The sentiment expressed by the target user's review of the album is used as a benchmark to find the set of users who have similar sentiments to those expressed by the target user's review of the same album, which can be viewed as the users in the set of users who have similar interests in the album as the target user. Users with similar interests are then found to predict the target user's rating of the unreviewed album through the nearest neighbor users.

II. B. 1) User similarity calculation

(1) Constructing the user-album rating matrix

Since this paper uses a polynomial plain Bayesian classifier for sentiment analysis, and the more complex the model, the easier it is to produce overfitting, in order to solve this problem, this chapter will be five ratings “Acclaim”, “Favorable”, “Mixed”, “Dislike”, and “Unfavorable” correspond to 90, 70, 50, 30, and 10 points respectively. Since reviews tend to give a more realistic representation of a user's actual rating, the rating R_g corresponding to a user's review corresponds to the user's scoring R_s with a weight coefficient of $\lambda:(1-\lambda)$. Assuming that R_{uc} denotes the rating score of user u for album c, here R_{uc} is defined as shown in Equation (7):

$$R_{uc} = \lambda R_g + (1-\lambda) R_s \tag{7}$$

The similarity between users is calculated by the ratings of two users for different albums. Each user is represented using an N-dimensional vector, and what each dimension represents is that user's rating of the album. The following equation (8) for user u is used:

$$U = \{R_{u1}, R_{u2}, R_{u3}, \cdots R_{un}\} \tag{8}$$

where there are a total of n albums in the training set, and $R_{u,1}$ denotes the rating of user U for the first album in the list of albums in the training set. All user rating vectors are formed into a user-album rating matrix, the representation of which is shown in Table 1.

Table 1: User-Album Rating Matrix

Name	c_1	c_2	c_3	$\cdots$	c_n
u_1	R_{11}	R_{12}	R_{13}	$\cdots$	R_{1n}
u_2	R_{21}	R_{22}	R_{23}	$\cdots$	R_{2n}
u_3	R_{31}	R_{32}	R_{33}	$\cdots$	R_{3n}
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
u_m	R_{m1}	R_{m2}	R_{m3}	$\cdots$	R_{mn}

(2) Calculating similarity

In most published papers, the use of Pearson's correlation coefficient tends to produce better similarity measures for recommendation accuracy in rating-based recommender systems. However, Pearson's correlation coefficient has a serious drawback when the data is sparse: in extreme cases, when there is only one review for some albums, it will be impossible to compute, and when both users have reviewed only one album, their similarity will be 1, which is often very inaccurate.

So there is a natural, parameter-free way to solve these problems, where Pearson correlation can be computed for all items when two users have very few reviews of the same album, and if the albums that users have not rated are treated as if they had the average rating of the users, rather than just being discarded. When computed using sparse vectors, this can be accomplished by subtracting each user's average rating from each user's rating vector, and then comparing the users by taking the cosine between their rating vectors and assuming that the missing value is zero. Equation (9) below:

$$sim(u,v) = \frac{\Sigma_{Cel_{uv}} (R_{uc} - \overline{R_u})(R_{vc} - \overline{R_v})}{\sqrt{\Sigma_{Cel_{uv}} (R_{uc} - \overline{R_u})^2} \sqrt{\Sigma_{Cel_{uv}} (R_{vc} - \overline{R_v})^2}} \tag{9}$$

where l_{uv} denotes the set of albums rated simultaneously by user u,v. $\overline{R}_u, \overline{R}_v$ denote the average score of user u,v in the rating set I_{uv}, respectively.

II. B. 2) User selection based on KNN algorithm

With a method for finding the similarity between users, the similarity between the target user and all other users needs to be found. There are many different methods of nearest neighbor user selection in previous studies, such as by random sampling, by randomly selecting the set of nearest neighbor users, which often improves the computational performance of the algorithm. However, it affects the prediction accuracy to a certain extent, and it is also possible to use all the nearest neighbor users for prediction, which can seriously affect the performance of the algorithm. Or a threshold is set and only users above that similarity are adopted as the nearest neighbor users for the target use, but in the case of sparse data, this approach leads to some of the target users who are very different from the public interest, and there are not enough ratings of the nearest neighbor users for the ratings prediction of the recommended songs. The user selection process based on the KNN algorithm is shown in Figure 1. The K Nearest Neighbor algorithm was used for the selection of nearest neighbor users, where the K most similar nearest neighbor users are selected and then the ratings of the target users for the new album are predicted from these K users [23]. Past evaluations have shown that using 20 to 60 most similar users avoids excessive computation, in that it does not affect the accuracy of the algorithm to a great extent and improves the performance of the algorithm. In addition, a threshold can be set appropriately to filter the nearest neighbor users, which can effectively avoid the noise introduced by low-similarity neighbors, but this will increase the negative impact of data sparsity.

The specific process is that after calculating the similarity of the users, the i users with the highest similarity are selected, and the set of these users is denoted by S_1. Next, the average similarity is derived by calculating the average similarity between the target user and the users in S_1, and the result of the calculation is set as the similarity threshold. Get all the users whose similarity is greater than the similarity threshold added to the user set S_1 to get a new user set S_2. Finally, a minimum value j of the user set S_2 is set up, and when the number of users is less than j, the remaining users in S_1 are added to S_2 in order of similarity. When the number of users is greater than or equal to j, S_2 is the final set of users' nearest neighbors.

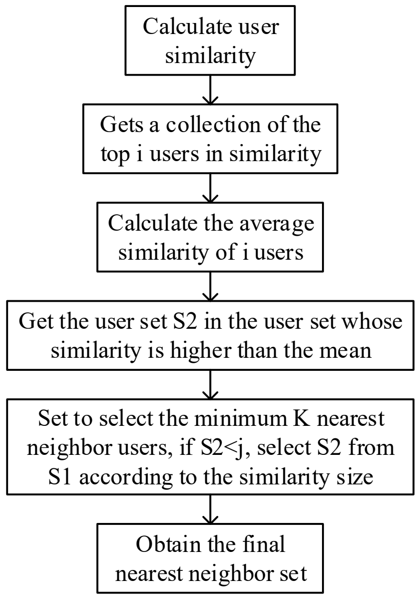


Figure 1: Flowchart of obtaining neighbor user sets

II. B. 3) Predictive Scoring and Recommendations

After the set of nearest neighbor users N_u is selected, the target user u can be given the rating prediction for the album c, and the predicted rating $S(u,i)$ is defined as shown in Equation (10):

$$S(u,c)=\frac{\Sigma_{v\in N_{uc}} sim(u,v)*R_{vc}}{\Sigma_{v\in N_{uc}} |sim(u,v)|} \tag{10}$$

where N_{uc} is a subset of the set of nearest neighbor users N_u, the set of users in the set of nearest neighbor users who have ratings for the album c. R_{vc} is the rating of user v for album c.

The basis of this prediction method in this chapter is also normalized to obtain a weighted average of the predicted album offsets from the average user ratings. In order to solve the effect of different users' different rating criteria and increase the comparability of the prediction results, as shown in Equation (11):

$$S(u,c)=\mu_u+\frac{\sum_{v\in N_{uc}} \sin(u,v)*(r_{vc}-\mu_v)}{\sum_{v\in N_{uc}} |\sin(u,v)|} \tag{11}$$

where μ_u, μ_v are the average ratings of the target user u and the nearest neighbor user v on all rated albums, respectively.

After obtaining the predicted ratings of the target user for all unrated albums, these predicted ratings are sorted and the N highest rated albums are taken as the target user's recommendations using the Top-N method.

III. Design of Music Course Content Recommendation System for Colleges and Universities

This chapter designs and develops a college music course content recommendation system based on the algorithm proposed in the previous section combined with Spring Boot, MyBatis, Vue and other technologies. The system allows students to search for music directly, and users can also choose to recommend their favorite music through the music course content recommendation module, so that students' listening experience can be enhanced.

III. A. Needs analysis

III. A. 1) Overview of requirements

Achieving accurate music course content recommendation is the main issue to be considered by the music course content recommendation system in terms of improving the student experience and the accuracy of the system recommendation. From the students' perspective, the system needs to recommend appropriate music course content to students based on their feedback and the system recommendation strategy. From the system's point of view, the system needs to determine whether to recommend the appropriate music to students based on the rating of the music. After analyzing the system requirements, the data flow diagram and business flow diagram are obtained, and the system is split into four modules, among which, the student information management module contains two roles of students and teachers, students can log in to the system after registering and modify their personal information, etc. Teachers are mainly responsible for managing students' information in this module.

III. A. 2) System role analysis

According to the business flow diagram of the system, the system is operated by teachers and students. Students can manage their personal information, select the music course content recommendation algorithm, search for music, and comment on music, etc. Teachers can manage student information and music course content information through this system.

III. A. 3) Functional requirements analysis

(1) Requirement analysis of student information management module

The function of this module is the information management function of two roles, students register and login on the system and edit their information. Teachers mainly operate in this module by viewing and editing students' information.

(2) Requirement Analysis of Music Course Content Recommendation Module

The function of this module is the music course content recommendation function, which includes the students' choice of recommendation algorithm and the result of music course content recommendation.

(3) Requirement analysis of music course information management module

This module is divided into the editing of music course information by teachers and the editing of music course comments by students.

(4) Requirement Analysis of Music Review/Search Module

This module is divided into the functions of students searching for relevant music course content through the search box, and entering the music course details interface to rate and comment on the music, etc. If there is no result in the search, students are prompted that the music course content does not exist.

III. A. 4) Analysis of non-functional requirements

System non-functional requirements refer to some special requirements that the software system must meet in addition to the realization of the function, in order to ensure that the system can run stably, reliably and efficiently. The first is performance requirements, including response time, throughput, resource utilization and so on. The system needs to complete the task within the specified time and ensure the efficient utilization of system resources. This system is verified through black-box testing that the functions of each module of the system keep running normally and the response time is fast. Next is the usability requirements, including the system is easy to use, easy to learn, easy to understand and so on. This system is more comprehensive and the system interface is intuitive and user-friendly, so that users can easily use the system and get started quickly. Lastly, maintainability and scalability requirements, including the system is easy to maintain, easy to modify, easy to upgrade.

III. B. System design

III. B. 1) Overall architectural design

In this paper, the system implementation used is Spring Boot + MyBatis + Vue framework.Spring Boot as a lightweight, simplified configuration based on the Spring Framework framework, in the development process has many advantages. It can automatically complete most of the project configuration, eliminating the tedious XML configuration, embedded Web server, so that developers can focus more on the implementation of business logic, greatly improving the development efficiency and development experience. In addition, Spring Boot also provides production-level monitoring, health checks, and external optimization configuration options to facilitate the developer in the production environment to optimize the application. MyBatis is a persistence layer framework for customizing SQL and stored procedures. It uses XML and annotations to configure native information, thus avoiding the hassle of using JDBC code and setting parameters manually.Vue focuses on the view layer and is particularly good at efficiently binding data to DOM structures, reducing developers' attention to underlying details and improving the efficiency of business logic implementation.

III. B. 2) Core functional module design

The music course content recommendation system module studied in this paper is mainly divided into four parts: student information management module, music course content recommendation module, music information management module, and music search comment module. Student information management module, the module is operated by students and teachers, students register on the system to log in and edit their own information, teachers in this module, the main mode of operation is to view and edit student information. Music course content recommendation module, the module realizes the function of selecting the recommended algorithm to recommend the music course content, students log in to the system through the different recommended methods selected, click on the recommended methods to return to the corresponding recommended results page. Music information management module, in this module, teachers can edit and manage music information and users' unreasonable comments on music. Music search and comment module, the module, students through the search box to search for the relevant music or singer, and enter the music details interface on the music ratings and comments, etc., if the search results do not prompt the teacher that the music does not exist.

III. B. 3) Database design

(1) System E-R diagram design

This system selects teacher ID, student ID, and music course content ID as the primary key, and designs the system E-R diagram based on the relationship between different entities.

(2) System database table design

By analyzing the requirements of the system, database tables were designed and constructed, including student information table, teacher information table, and music course content information table.

IV. Empirical Analysis of Music Course Content Recommendation in Colleges and Universities

IV. A. Empirical Analysis of Sentiment Analysis Algorithms

IV. A. 1) Experimental environment and data

(1) Experimental environment

Hardware environment: this experiment was conducted on a personal PC with an Intel(R)Core(TM)i7-4720HQ CPU@2.60GHz processor and 16GB of RAM.

Software environment: the operating system is Windows 10, and the development language is Python 3.5, based on the deep learning framework Keras. Keras is an open-source artificial neural network library written in Python, which can be used as a high-level application program interface to Tensorflow, Microsoft-CNTK, and Theano for deep learning model design, debugging, evaluation, application, and visualization.Keras can be prototyped easily and quickly, supports both convolutional networks and recurrent networks, as well as combinations of the two, and supports seamless switching between CPUs and GPUs.

(2) Experimental data

The experimental data comes from music platforms such as NetEase Cloud Music, QQ Music, etc. The song list with emotion tags is crawled through the crawler technology, and then the lyrics of the corresponding music courses are crawled, by this way, the teaching content of music courses with emotion tags can be obtained, and then the processed lyrics are deposited into the database, and the classification of the processed experimental data is shown in Table 2. Meanwhile, for this dataset, 80% is selected as the training set and 20% as the testing set.

Table 2: Experimental data classification

Name	Emotional color	Number of Music
1	Sorrow	418
2	Happy	558
3	Calmness	619
4	Excited	676

After preprocessing the dataset such as removing deactivated words, the distribution of the number of words of the lyrics needs to be counted before vectorization of the lyrics, so that the optimal sentence length can be selected. Because the input dimension of the neural network is fixed, it is necessary to select the appropriate sentence length, cut off the excessively long sentences, and fill in the insufficient part, and the distribution of the number of words in the text after preprocessing is shown in Figure 2. The number of words in the processed lyrics is basically distributed in the range of 0-75, so in the experiment, the sentence length is set to 75, which ensures that most of the songs are covered, and that the length is not too large or too small, which makes the filling of too many zeros or too much interception of the sentence, thus resulting in deviation of the classification results.

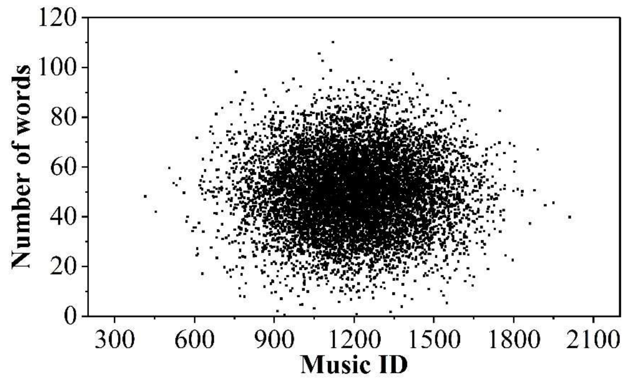


Figure 2: Distribution of the number of text words after preprocessing

IV. A. 2) Experimental program

In order to verify the effectiveness of the sentiment analysis method proposed in this chapter, this paper will compare with LSTM model, BiLSTM model, TextCNN model and the method proposed in this paper under the same experimental environment and using the same experimental data. For text categorization models, the confusion matrix is usually used to calculate the common evaluation metrics such as accuracy, precision, recall and F1 value.P and N denote the positivity and negativity of the prediction results, and T and N denote the consistency between the actual results and the predicted results, and the consistency is true and the inconsistency is false. Given the confusion matrix, the calculation of accuracy, precision, recall and F1 value are shown in Eqs. (12), (13) (14) (15), respectively:

$$Accuracy = \frac{TP+TN}{TP+TN+FN+FP} \tag{12}$$

$$Precision = \frac{TP}{TP+FP} \tag{13}$$

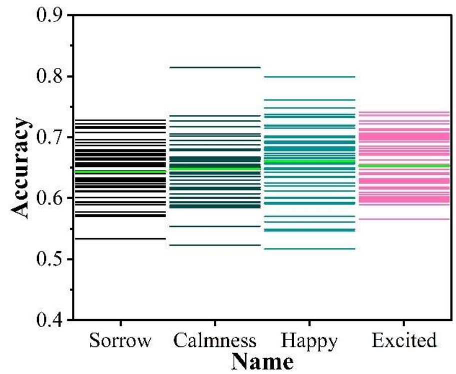
$$Recall = \frac{TP}{TP+FN} \tag{14}$$

$$F1 = \frac{2*\Pr ecision*\operatorname{Re} call}{\Pr ecision+\operatorname{Re} call} \tag{15}$$

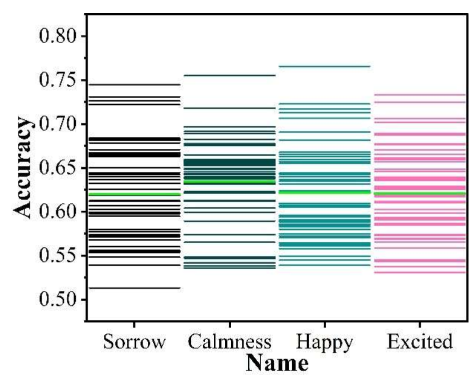
After comprehensive consideration, this paper chooses accuracy rate and F1 value as the evaluation indexes of the model in this paper.

IV. A. 3) Experimental results and analysis

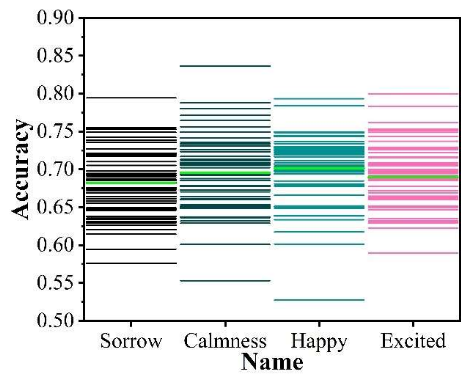
In the experimental process, it is necessary to use some fixed parameters, and the values of these parameters will directly affect the experimental results, so it is more important to choose reasonable parameters. Under the same experimental conditions, the accuracy and F1 value of different models are verified. Where BiLSTM stands for bidirectional long and short-term memory network, LSTM stands for long and short-term memory network, Text CNN is a convolutional neural network model, and all the above models are classical emotion classification models, and the advantages and disadvantages of the method proposed in this paper can be verified by comparing with these models. In the following experimental comparison, the classification method proposed by the text is named Ours for differentiation. Sentiment analysis accuracy comparison is shown in Fig. 3, and sentiment analysis F1 value comparison is shown in Fig. 4, where (a) to (d) are BiLSTM, LSTM, Text CNN, and Ours, respectively. from the results of accuracy, the values of the methods proposed in this paper are all the highest, and textcnn is also higher than bilstm and lstm, which indicates that in music sentiment classification, the use of the polynomial plain Bayesian classifier can better discover the emotion features in music and thus obtain better classification results. From the results of F1 value, the F1 value of this paper's method is still the highest relative to other models, but it does not exceed 75% on the sad, calm, sorrowful, and agitated categories, which, in combination with the performance of these four categories of emotional colors in terms of accuracy, can be considered as a situation where some positive samples in the dataset are labeled as negative samples. In this case, the model becomes weaker in learning for positive samples, such as sad type of songs, which ultimately leads to low recall and makes the F1 value low. Meanwhile, from the sample size of the experimental data, the sample size of the four emotion types is evaluated at less than 1,000 items, and this amount of data may lead to insufficient feature values learned by the model, resulting in the F1 value not reaching a high level. In the subsequent experiments, increasing the sample size and re-verifying the classification labels can be considered to improve the final classification effect.



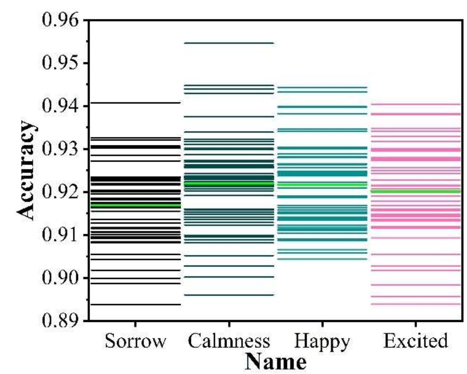
(a)BiLSTM



(b)LSTM

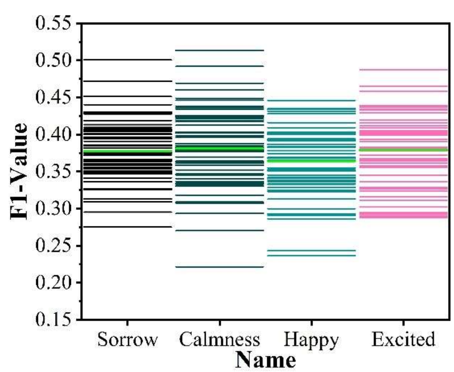


(c)Text CNN

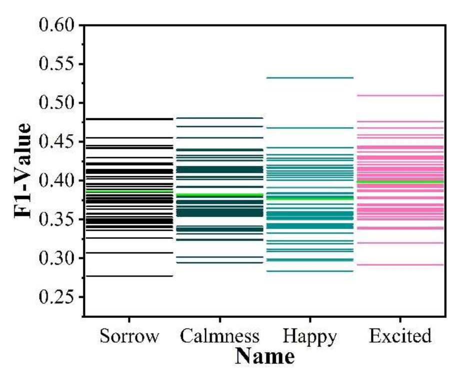


(d)Ours

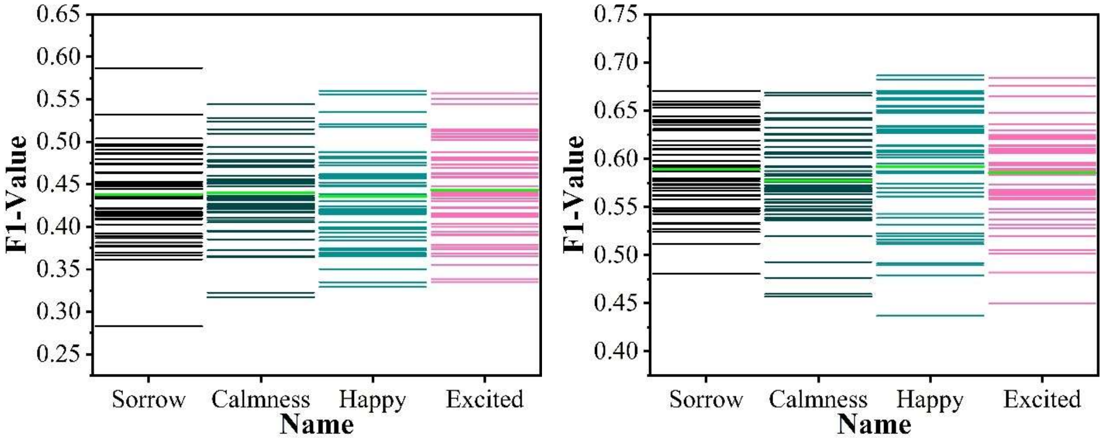
Figure 3: Accuracy comparison of sentiment analysis



(a)BiLSTM



(b)LSTM



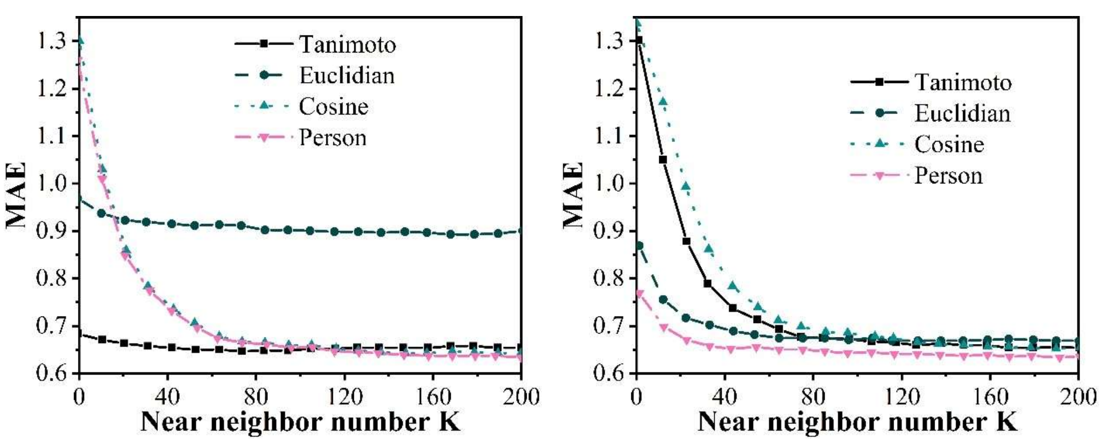
(c)Text CNN (d)Ours

Figure 4: Comparison of F1 values in sentiment analysis

IV. B. Empirical Analysis of Recommendation Algorithms

IV. B. 1) Comparative analysis of similarity

KNN algorithms with different similarity metrics are considered to compare the performance: KNN-Pearson, KNN-Cosine, KNN-Euclidian, and KNN-Tanimoto.The first experiment is done to find the optimal value of the number of nearest neighbors K in the dataset and incremental dataset.The KNN algorithms with different similarity metrics The results of the comparison of MAE values are shown in Fig. 5, where (a)~(b) are the test set and validation set, respectively. In order to determine the optimal value of the number of nearest neighbors K for the KNN method on the test set and validation set, the experimental selection range is controlled between 10 and 200. By increasing the value of K it was found that the KNN algorithm using different similarity measures gave better prediction accuracy. In Fig. 5(a), the results of MAE values using KNN-Pearson and KNN-Cosine with varying number of nearest neighbors are extremely similar on the test set, with KNN-Pearson outperforming KNN-Tanimoto when K < 100 and when K > 100. In conclusion, it can be seen from Fig. 5 that overall, by varying the number of nearest neighbors, it can be found that KNN-Pearson performs better than other KNN algorithms on the test and validation sets.

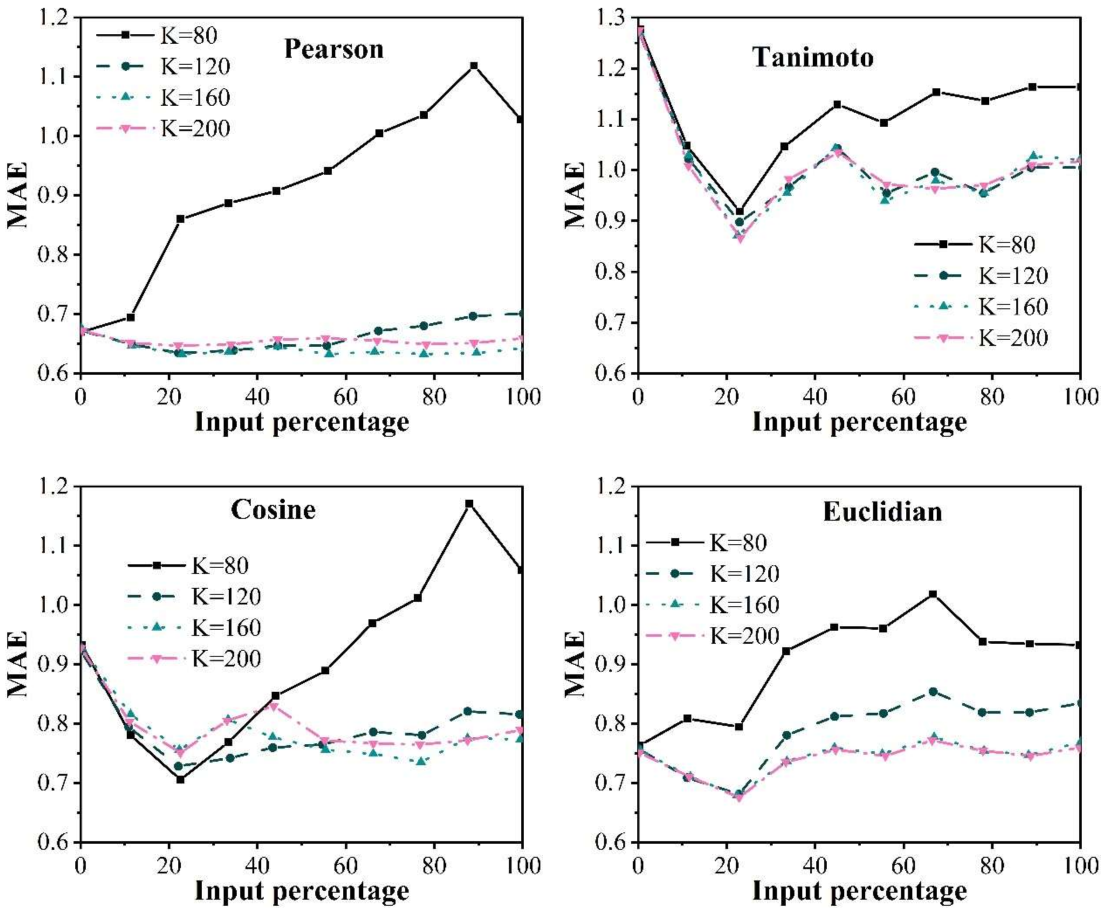


(a)Test set (b)Validation set

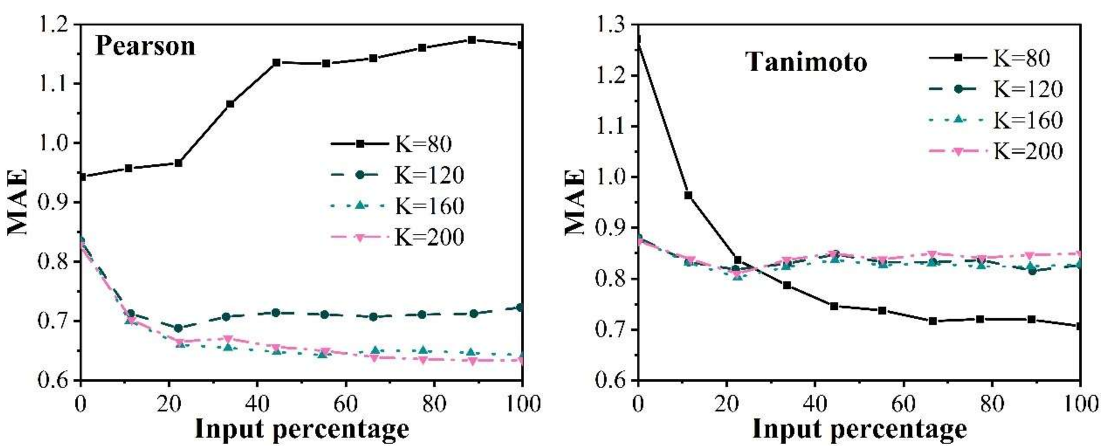
Figure 5: MAE values of KNN algorithms with different similarity measures

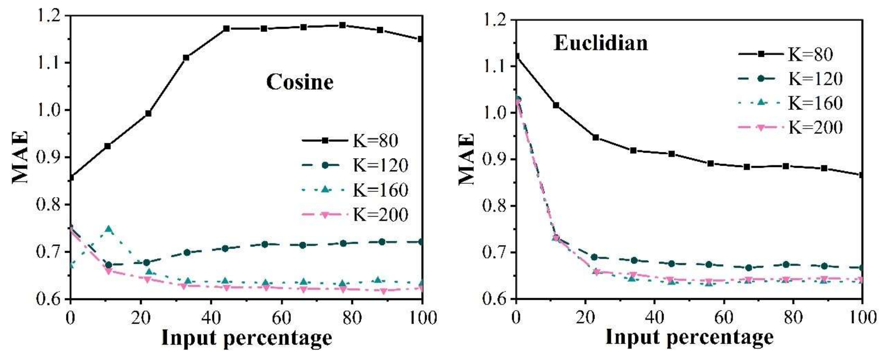
Experimentation by K values set to 20, 60, 150 and 190 shows that the KNN algorithm for most of the similarity metrics, except for K = 20, obtains the optimal prediction with the increase in the number of users with different K values, and since the MAE value corresponding to K = 150 is the smallest when K=150 corresponds to the smallest MAE value, then it is the optimal value, and the comparative similarity analysis is shown in Figure 6. The above experimental results show that the Pearson similarity measure performs better. In Fig. 6(a), for the test set, its performance is optimal when $K \approx$ 160. In Fig. 6(b), for the validation set, its performance is optimal when $K \approx$

200. Based on the above results the K value is used and the accuracy of the proposed algorithm is further verified in the next experiments, not only considering the KNN algorithm with different similarity measures, but also adding the K-means algorithm and Baseline algorithm, and the results of the similarity experiments are shown in Fig. 7. In Fig. 7(a), KNN-Pearson performs better in any case in the test set. In Fig. 7(b), the K-Means algorithm outperforms the other algorithms when the percentage of incremental data input to the training set is between 10% and 30%. When the percentage is more than 35%, the MAE value of KNN-Pearson is the smallest, which indicates that the Pearson coefficient similarity formula used in this paper is more conducive to the work of recommending the content of music courses in colleges and universities.



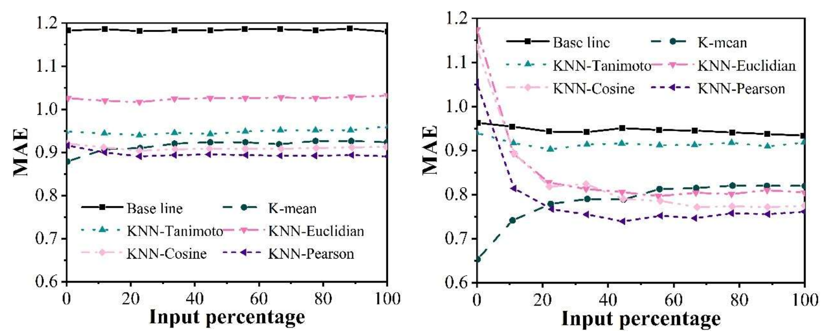
(a)Test set





(b)Validation set

Figure 6: Comparative analysis of similarity



(a)Test set (b)Validation set

Figure 7: The result of similarity experiment.

IV. B. 2) Comparative analysis of the effectiveness of recommendations

In order to observe the effect of K value on the music course content recommendation effect, so this comparison experiment was done. Let this paper's algorithm be compared with other recommendation algorithms containing nearest neighbors, the results of the comparative analysis of the music course content recommendation effect are shown in Figure 8. Compare this paper's algorithm with KNN, KNN-DR (a hybrid method combining matrix decomposition and nearest neighbor algorithm), and KNN-PCA algorithm, which is a hybrid method combining the principal component analysis method with the nearest neighbor algorithm, and does the recommendation through the collaborative filtering algorithm of dimensionality reduction. The horizontal axis is the K value taken, which is taken when K=20, 40, 60, 80, 100, 120, 140, 160, 180, 200 to do the calculation, and the vertical axis is the Accuracy value of the effect of music course content recommendation. From the figure, it can be seen that for the same K value, the music course content recommendation accuracy of this paper's algorithm is the best, and its value is maintained above 0.9, which indicates that the introduction of Pearson similarity and polynomial plain Bayesian classifier on the basis of KNN algorithm can effectively improve the quality of course content recommendation in colleges and universities, so as to make it better to serve students and teachers.

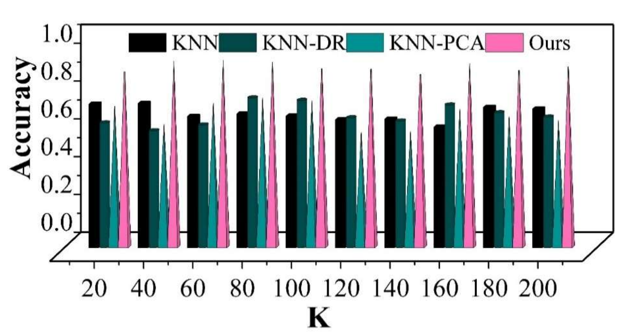


Figure 8: Comparison of music course content recommendation effect

IV. C. Empirical Analysis of Recommendation Systems

IV. C. 1) Testing tools

In the experiment, the web automation testing tool chosen to simplify the testing is Watir. The Watir framework is implemented through Ruby language.Watir comes with rich tool functions and is an open source, outstanding testing tool and also Watir contains tools for recording test results.

IV. C. 2) Performance evaluation indicators

In the test, two evaluation metrics are used to evaluate the recommended performance of the design system, the two metrics are interest fit, shikikai fit, and MAP i.e. Mean Average Precision where the formula for the interest fit is specified as follows:

$$S_P = \frac{2\left(\sum_{1}^{n}(m_p - d_i)\right)}{n} \tag{16}$$

where n is the number of times the user has previously downloaded songs. m_p is the category of songs preferred by the user obtained from the analysis. d, is the category of songs previously downloaded by the user. The formula for $m_p - d_i$ is specified as follows:

$$m_p - d_i = \begin{cases} 1 & m_p = d_i \\ 0 & m_p \neq d_i \end{cases} \tag{17}$$

The formula for calculating scenario fit is specified below:

$$S_C = \frac{2\left(\sum_{1}^{n}(m_c - c_i)\right)}{n} \tag{18}$$

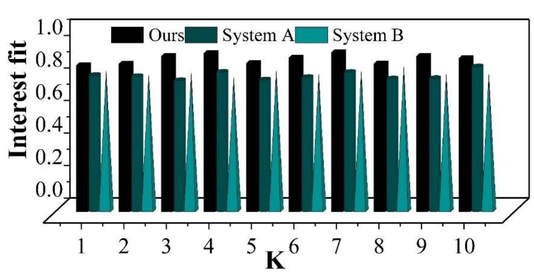
where m_c is the user preference shy scenario obtained from the analysis. c_i is the scenario of the user's previous song downloads. The formula for $m_c - c_i$ is specified as follows:

$$m_c - c_i = \begin{cases} 1 & m_c = c_i \\ 0 & m_c \neq c_i \end{cases} \tag{19}$$

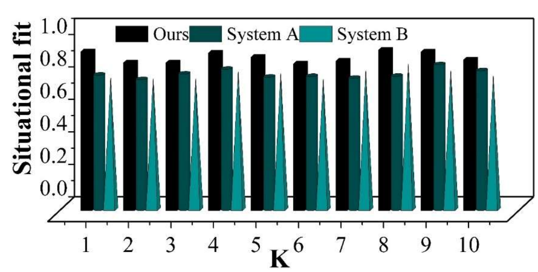
The test of Mean Average Precision (MAP) includes Download Mean Average Precision (MAP) and Collection Mean Average Precision (MAP).The higher the value of MAP indicator, the better the recommendation performance of the system. The real-time update delay, real-time recommendation delay, and throughput are used as evaluation indexes for the system operation performance test.

IV. C. 3) Analysis of Recommended Performance Test Results

Firstly, we test the recommendation performance of the system, that is, we test the interest match and scenario match of the system, and the specific test results are shown in Fig. 9, where (a) ~ (b) interest match and scenario match. According to the test data of system interest coincidence and scenario coincidence in the figure, when the design system is used to recommend music course content to 10 users, the interest coincidence and scenario coincidence of the recommendation results of the design system in this paper are high, and the interest coincidence degree can reach 97.816% and the scenario coincidence degree can reach 98.625%, which is much higher than that of the other two systems (A and B), indicating that the recommendation performance of the design system in this paper is better.



(a)Interest fit



(b)Situational fit

Figure 9: The test results of the system's interest fit and situation fit

In the test of the average precision mean value of downloads and the average precision mean value of collections of the system, the historical behavior logs of users before a certain time node are taken as known logs, and the historical behavior logs of users after that time node are taken as test logs. From this node, the system generates a list of recommended tracks to the user, and compares the test log with the generated list of recommended tracks based on the time node, and tests the average precision mean value of downloads and the average precision mean value of favorites at each time node.The average test results of 10 users are shown in Fig. 10. The user average test results in Fig. 10 show that the system's download average precision mean and collection average precision mean are high, indicating that the system's recommendation performance is good.

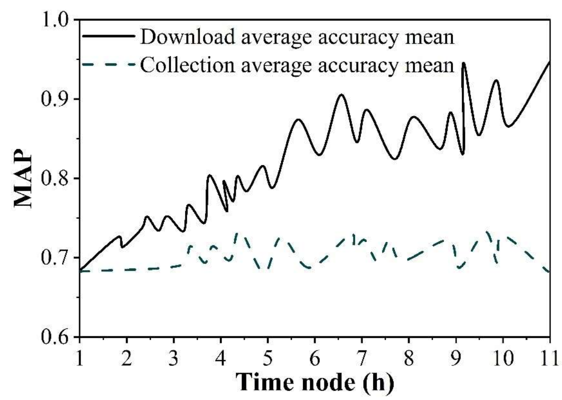
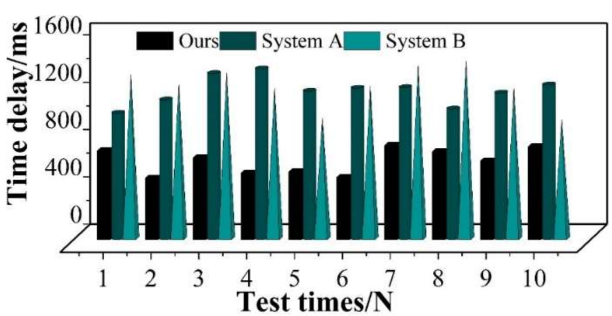


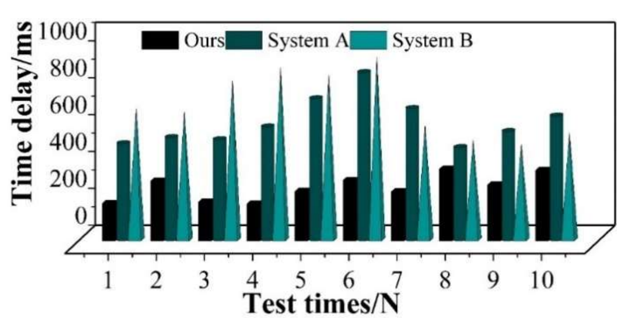
Figure 10: Average test results for 10 users

IV. C. 4) Analysis of system operation performance test results

Next, the operational performance of the system is tested, that is, the real-time update delay and real-time recommendation delay are tested when the system is running, and the results of the system operational performance test are shown in Fig. 11, where (a)~(b) are the real-time update delay and real-time recommendation delay, respectively. According to the test data of real-time update delay and real-time recommendation delay in Fig. 11, the real-time update delay of the designed system is shorter than 1000ms, and the real-time recommendation delay is shorter than 500ms, which is much lower than that of the other two systems, which proves that the system designed in this paper has a good performance, can meet the needs of students and teachers in colleges and universities, and has a facilitating role in the development of the digitization of music courses in colleges and universities.



(a)Real-time update delay



(b)Real-time recommendation delay

Figure 11: Real-time update delay and real-time recommended delay test results

V. Conclusion

In this paper, we obtain the data of this research with the help of web crawler technology and preprocess the data of this research. Using TF-IDF to complete the work of sentiment feature extraction, the features are put into the polynomial plain Bayesian network for iterative training, and finally complete the task of constructing the sentiment analysis algorithm based on user comments. After obtaining the sentiment classification results of user comments, Pearson similarity algorithm and K nearest neighbor algorithm are used to jointly realize the music course content recommendation algorithm design project. Combining the above algorithms and related program development software, the music course content recommendation system for colleges and universities is designed and implemented, and the operational performance performance of the system is explored. The interest matching degree and scenario matching degree of this paper's system are kept above 90%, which is much better than the other two control systems, verifying that this paper's recommender system can meet the needs of students and teachers, and has a guiding value for the development and construction of music courses in colleges and universities.

References

[1] Henninger, J. (2018). to-resource: Effective incorporation of world music into the music classroom. Update: Applications of Research in Music Education, 37(1), 5-8.

[2] Liu, D., Lin, X., Li, L., & Ming, Z. (2024). Teaching content recommendations in music appreciation courses via graph embedding learning. International Journal of Machine Learning and Cybernetics, 15(9), 3847-3862.

[3] Caamaño Liñares, T., Rodríguez Rodríguez, J., Castro Rodríguez, M., & Marín Suelves, D. (2023). Digital didactic resources and music: mapping the last decade of research. Music Education Research, 25(4), 351-366.

[4] Stefanova, P., Stefanov, P., & Doychinov, Y. (2021). Creating digital educational content-opportunities and perspectives for creative interaction in music education. In EDULEARN21 Proceedings (pp. 4606-4611). IATED.

[5] García, I. D., Acero, J. M. A., de las Heras-Fernández, R., & Calderón-Garrido, D. (2021). Digital competence and the use of technological resources by teachers in music conservatories and schools of music. Música Hodie, 21.

[6] Velankar, M., & Kulkarni, P. (2022). Music recommendation systems: overview and challenges. Advances in Speech and Music Technology: Computational Aspects and Applications, 51-69.

[7] Schedl, M., Zamani, H., Chen, C. W., Deldjoo, Y., & Elahi, M. (2018). Current challenges and visions in music recommender systems research. International Journal of Multimedia Information Retrieval, 7, 95-116.

[8] Amrutha, B., & Supriya, M. (2023, July). Recommendation of Independent Music based on Sentiment Analysis. In 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT) (pp. 1-6). IEEE.

[9] Satornicio Medina, A. L., Sucari León, R., & Calderón-Vilca, H. D. (2023). Music recommender system based on sentiment analysis enhanced with natural language processing technics. Computación y Sistemas, 27(1), 53-62.

[10] Dang, C. N., Moreno-García, M. N., & Prieta, F. D. L. (2021). An approach to integrating sentiment analysis into recommender systems. Sensors, 21(16), 5666.

[11] Keluskar, S. S., Dhuri, V. L., Gonjari, S. S., & Sanghavi, N. (2022). Mehfil: song recommendation system using sentiment detected. Int J Eng Res & Technol (IJERT).

[12] Wani, V., Bothe, N., & Soni, A. (2021). Music Suggestion Via Sentimental Analysis of User-Inputted Texts. International Journal of Scientific Research in Computer Science Engineering and Information Technology.

[13] Moscato, V., Picariello, A., & Sperli, G. (2020). An emotional recommender system for music. IEEE Intelligent Systems, 36(5), 57-68.

[14] Rosa, R. L., Rodriguez, D. Z., & Bressan, G. (2015). Music recommendation system based on user's sentiments extracted from social networks. IEEE Transactions on Consumer Electronics, 61(3), 359-367.

[15] Lee, S. J., Seo, B. G., & Park, D. H. (2018). Development of music recommendation system based on customer sentiment analysis. Journal of Intelligence and Information Systems, 24(4), 197-217.

[16] Akuma, S., Obilikwu, P., & Ahar, E. (2021). Sentiment analysis of social media content for music recommendation. NIGERIAN ANNALS OF PURE AND APPLIED SCIENCES, 4(1), 95-102.

[17] Wang, D. (2022). Analysis of sentiment and personalised recommendation in musical performance. Computational Intelligence and Neuroscience, 2022(1), 2778181.

[18] Ayata, D., Yaslan, Y., & Kamasak, M. E. (2018). Emotion based music recommendation system using wearable physiological sensors. IEEE transactions on consumer electronics, 64(2), 196-203.

[19] Vidyasagar, B. S., & Karwande, V. (2021). Emotion based music recommendation system by using different ML approach. International Journal, 6(6).

[20] Nguyen, H., Tran, N., Ly, D., Tran, A., Nguyen, A., & Vo, H. (2024). A Model for Song Recommendation Based on Facial Emotion Analysis and Musical Emotion. International Journal of Intelligent Engineering & Systems, 17(4).

[21] Wantong Yang,Enze Wang,Zhiwen Gui,Yuan Zhou,Baosheng Wang & Wei Xie. (2025). An MLLM-Assisted Web Crawler Approach for Web Application Fuzzing. Applied Sciences,15(2),962-962.

[22] C.A Nurhaliza Agustina,Rice Novita,Mustakim & Nesdi Evrilyan Rozanda. (2024). The Implementation of TF-IDF and Word2Vec on Booster Vaccine Sentiment Analysis Using Support Vector Machine Algorithm. Procedia Computer Science,234,156-163.

[23] Himanshi Babbar,Shalli Rani,Dipak Kumar Sah,Salman A. AlQahtani & Ali Kashif Bashir. (2023). Detection of Android Malware in the Internet of Things through the K-Nearest Neighbor Algorithm. Sensors,23(16).