sustainability

Article

Artificial Intelligence Classification Model for Modern Chinese Poetry in Educatio

Mini Zhu [1], Gang Wang [2], Chaoping Li [1,*], Hongjun Wang [2,*] and Bin Zhang [3]

1 College of Arts, Chongqing Three Gorges University, Chongqing 404188, China
2 School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu 611756, China
3 Shanghai Film Academy, Shanghai University, Shanghai 200444, China
* Correspondence: chaopingli00@gmail.com (C.L.); wanghongjun@swjtu.edu.cn (H.W.)

Abstract: Various modern Chinese poetry styles have influenced the development of new Chinese poetry; therefore, the classification of poetry style is very important for understanding these poems and promoting education regarding new Chinese poetry. For poetry learners, due to a lack of experience, it is difficult to accurately judge the style of poetry, which makes it difficult for learners to understand poetry. For poetry researchers, classification of poetry styles in modern poetry is mainly carried out by experts, and there are some disputes between them, which leads to the incorrect and subjective classification of modern poetry. To solve these problems in the classification of modern Chinese poetry, the eXtreme Gradient Boosting (XGBoost) algorithm is used in this paper to build an automatic classification model of modern Chinese poetry, which can automatically and objectively classify poetry. First, modern Chinese poetry is divided into words, and stopwords are removed. Then, Doc2Vec is used to obtain the vector of each poem. The classification model for modern Chinese poetry was iteratively trained using XGBoost, and each iteration promotes the optimization of the next generation of the model until the automatic classification model of modern Chinese poetry is obtained, which is named Modern Chinese Poetry based on XGBoost (XGBoost-MCP). Finally, the XGBoost-MCP model built in this paper was used in experiments on real datasets and compared with Support Vector Machine (SVM), Deep Neural Network (DNN), and Decision Tree (DT) models. The experimental results show that the XGBoost-MCP model performs above 90% in all data evaluations, is obviously superior to the other three algorithms, and has high accuracy and objectivity. Applying this to education can help learners and researchers better understand and study poetry.

Keywords: XGBoost; poetry style classification; Chinese modern poetry; education

Citation: Zhu, M.; Wang, G.; Li, C.; Wang, H.; Zhang, B. Artificial Intelligence Classification Model for Modern Chinese Poetry in Education. *Sustainability* **2023**, *15*, 5265. https://doi.org/10.3390/su15065265

Academic Editors: Xuesong (Andy) Gao and Hao-Chiang Koong Lin

Received: 29 January 2023
Revised: 9 March 2023
Accepted: 10 March 2023
Published: 16 March 2023

1. Introduction

The study of literary works from a computational perspective can be traced back to the use of quantitative methods to record the works' lexical features and analyze the author's writing style. With the development of natural language processing techniques, automatic text classification offers a new approach to the application of computers to literary research; for example, Christou and Tsoumakas proposed a deep-learning-based approach to discovering semantic relationships in literary texts, and this effectively deals with the specificity of literary texts [1]. Much of the research on classification technology has also provided important technical support for the task of literary classification [2–4]. These developments play a very important methodological role in the study of the themes, authors, and styles of literary works. Research on the classification of literature is currently being conducted in various literary genres, such as poetry, prose and fiction. This research has focused on the identification of authorship, classification of textual styles, and analysis of textual emotions. In recent years, a number of scholars have conducted research on the automatic classification of poetry. These studies have been conducted in numerous languages, including English [5–8], Persian [9], Ottoman [10], Malaysian [11], Spanish [12], Arabic [13], Bengali [14], Punjabi [15], and Chinese [16–19].

In the study of modern new Chinese poetry, Voigt and Jurafsky discussed the extent to which the formal language of classical Chinese poetry was stripped away with the advent of Chinese vernacular poetry and used natural language processing techniques to analyze whether contemporary poetry is still connected to classical poetry. They succeeded in tracing the declining influence of classical poetic style and language in twentieth-century Chinese poetry by calculating three characteristics. This study provides a better understanding of the development of Chinese poetry [19].

The studies mentioned above all focus on the use of computers in literature. Based on these studies, this article further proposes the application of computers to literature education. The digitization of texts, along with the breakthroughs in artificial intelligence and natural language processing, have made it possible to conduct several studies on computer-assisted learning in the big data era [20–22]. Chang et al. recommended the use of automatic text categorization methods to categorize educational publications, which helps with educational innovation and serves as a useful source for this essay on the integration of artificial intelligence into the education of literature [23].

To accurately assess a poem's style for the purposes of the subsequent understanding and learning of the poem, students in traditional literary poetry education must acquire a significant amount of learning experience. Studies of the poets themselves and the literary history of poetry are two examples of literary studies of poetry that must be founded on an accurate assessment of poetic style. At present, the classification of modern poetry styles is mainly carried out by experts, which is inaccurate and subjective, and it is not conducive to the further study of literature. To make up for the shortcomings of traditional literature education, in this article, which aims to save students the time and effort of gaining experience, to more precisely assess the style of poetry, to increase the effectiveness of poetry learning, and to give literary researchers a more accurate reference on the subject of poetry classification for their upcoming academic work, an attempt is made to establish an automatic poetry classification model using text classification methods in natural language processing to provide an interactive artificial-intelligence-based learning environment for poetry learners and researchers. This is an innovative approach to traditional literary poetry education, providing new ideas for the development of poetry education and promoting its sustainability.

At present, research on the automatic classification of styles of new modern Chinese poetry has yet to be expanded. In this article, an attempt is made to establish an automatic poetry classification model using the text classification method in natural language processing to assist with poetry learning and research.

The main contributions of this paper are as follows.

- The Modern Chinese Poetry based on eXtreme Gradient Boosting (XGBoost-MCP) model is built in this paper, which makes poetry classification more accurate, objective, and efficient. It also resolves the defects in traditional poetry education.
- After labeling, word segmentation, and the removal of stopwords from the text data of modern Chinese poetry, the XGBoost-MCP model was iteratively trained using Doc2Vec and XGBoost algorithms, and, finally, we obtained the optimal model.
- Experiments on real datasets show that the XGBoost-MCP model is obviously superior to the Support Vector Machine (SVM), Deep Neural Network (DNN), and Decision Tree (DT) models, has certain accuracy and objectivity, and provides convenient auxiliary tools for poetry learners and researchers, which can promote sustainable poetry education.

Section 2 introduces related work, including the study of short text classification, literary classification, and current research on the classification of poetry in different languages, which has informed the research in this paper. Section 3 introduces the methods used to classify poetic styles and proposes a framework that is applicable to the classification of modern Chinese poetry. The concrete application of the XGBoost-MCP model in poetry education is also introduced. Section 4 introduces the experimental setup and evaluation indices and verifies the performance of the modern Chinese poetry classification model through the experimental results. Section 5 summarizes the full text and presents future work.

2. Related Works

Research on the application of natural language processing can be divided into classifications of topic [23–26], emotion [27,28], authorship, and style. The poetry text is short in form and literary in content. Therefore, research on short texts and literary text classification has an important reference value for this paper. In addition to the above, much of the research specifically evaluated and compared classification models.

2.1. Classification of Short Text and Literary Text

At present, short text classification is mainly focused on daily language use, such as text messages and forum questions and answers published by people on social media platforms.

Li et al. applied the pre-training method to a deep neural network based on a restricted Boltzmann machine to obtain a superior classification performance regarding user sentiment in short texts [29]. Rao et al. proposed a topic-level maximum entropy model for the socio-emotional classification of short texts and validated the model on real data [30]. Zheng et al. proposed the concept of sentiment context to improve performance in the context-based sentiment classification of short texts [31]. Liang et al. proposed a universal affective model, which consists of topic-level and term-level sub-models to classify readers' emotions using unlabeled short texts [32]. Chen et al. proposed a new method for emotion classification using complementary information for negative and intense emotions [33].

These studies on the emotional classification of short texts provide an important reference for the classification of poetry. Most studies of literary classification focus on the issue of authorship. The identification of authorship is essentially a determination of the author's style.

In 1963, Mosteller and Wallace proposed the automatic identification of authorship using the frequency and distribution of the occurrence of a few special words as features. Since then, a method of quantifying the stylistic features of texts has evolved to determine the style and authorship of literary texts [34]. Following closely behind Mosteller and Wallace, Holmes examined the topic of quantifying literary style. The feasibility of quantifying literary style traits to judge authors was discussed from multiple perspectives. He specifically examines the authors of biblical texts and provides an outlook on artificial neural networks' application to the analysis of literary texts [35]. Forsyth and Holmes built on their research into the quantification of the stylistic features of Holmes' literature in 1994, compared five methods of analysis of text features, and then tested them on ten representative text classification problems. The results showed that Monte Carlo feature-finding is more advantageous in the analysis of text features [36]. Argamon et al. expanded on the work of Mosteller and Wallace, as well as Holmes, in an effort to find a computable representation of linguistic features based on a systematic functional grammar. Their study illustrates how functional lexical features can play a role in the classification of various genres [37].

Based on studies of quantitative textual style features, researchers have further investigated automatic author identification, for example, by comparing writing styles to determine the gender of writers.

Koppel et al. built on the previous research and proposed the use of automatic text classification techniques in an attempt to infer the gender of authors using the latest automatic text classification methods such as K-nearest Neighbor (KNN), Neural Networks, Winnow, and SVM for simple combinations of lexical and syntactic features with an accuracy of 80% [38]. Argamon et al., following the research by Koppel et al., used an English corpus to examine the types of textual elements that were most successful in establishing automatic gender judgments for male and female authors [39]. Walkowiak and Piasecki extended the features based on grammatical classes, statistical features, and common words. They applied the Bow and FastText algorithms to examine various techniques for extracting text representation features in gender and author recognition stylometric tasks [40].

2.2. Classification Methods

Following the above study on the automatic identification of style, researchers have mostly quantified textual style features by capturing lexical items. This method was first achieved by manual metering. Later, with the development of natural-language-processing techniques, researchers applied text classifiers to assist in classification, which greatly improved the efficiency and accuracy of classification and facilitated the development of computer applications for literary research.

The framework for this field was established using the Naive Bayes (NB) classifier developed by Mosteller and Wallace [34].

Burrows originally conducted a Principal Component Analysis (PCA) of the frequency of function words and found similarities between the styles of different authors through projected graphs in 1987 [41]. Based on PCA, Gerard et al. introduced cluster analysis and discriminant analysis to the discipline [42]. Juola and Bayyen used the cross-entropy classification method to complement the deficiencies in past research using PCA and the discriminant analysis methods embodied in text classification. Experiments have shown that the technique can reliably detect authors as well as subtle features such as language family, topic, and even the education level of the author [17].

In 1999, Craig employed discriminant analysis to determine the classification of Middleton's problematic theatrical texts, and then used the results of his tests to describe Middleton's textual style [43].

Diederich et al. introduced this field to SVM. They tried to identify the articles of seven target authors from a group of 2652 news articles, written by multiple authors, covering three topical subjects in 2000 [44]. De Vel et al. used SVM to classify 150 email files by three authors, with an accuracy of 80% [45].

As classification systems have become more diverse, many studies have emerged to evaluate them. Zheng et al. extracted four writing style features and built a classification model based on the four styles using inductive learning algorithms. They compared the discriminative power of three classification techniques: DT, Back-propagation Neural Network, and SVM. Experimental accuracies of 70%–95% were achieved [46]. Yu compared the performance of NB and SVM classification algorithms in a literary classification task. Yu combined these two algorithms with three text-preprocessing tools to study the impact of these preprocessing tools on text-classification algorithms. The experimental results showed that SVM is usually better than NB. Yu argues that stopwords should be handled with caution. In Dickinson's poetry, stopwords are highly discriminative features, and their deletion would affect classification accuracy [47]. Mu classified Song dynasty poetry types using KNN, NB, and SVM, and discovered that SVM was the most successful, with an accuracy rate of above 95% [48]. Then, Pal and Patel utilized the bag-of-words model to extract Hindi text features and translate them into numerical representations, which were then transferred to the training model. The Hindi text was classified using KNN and NB to test the accuracy of the algorithms [49]. Kalcheva et al. compared the precision and system performance of machine learning algorithms to categorize texts produced by Bulgarian authors. The studied algorithms were NB, SVM, Random Forest, and AdaBoost. The results demonstrated that NB was the most precise and quickest algorithm for categorizing Bulgarian texts authored by two writers with the same number of poems [50].

Recent studies have merged artificial intelligence's deep learning techniques to enhance earlier classification models. Ahmad et al. suggested using deep learning, which is the most cutting-edge artificial intelligence technology, to categorize the emotions of poetry. A poetry corpus was used to create an Attention-Based Convolutional Bidirectional Long Short-Term Memory model. They divided poems into categories based on various emotional states, such as love, happiness, hope, grief, and wrath. In the experiment, they compared the established model with other machine learning models. The experimental results showed that their model has 88% accuracy [28]. Wei et al. suggested an improved K-means clustering algorithm for the classification of literary vocabulary [51]. Based on Ahmad's research, in 2022, Khattak et al. suggested an emotion categorization method for poetry texts using a DNN model. A Bidirectional Long Short-Term Memory model was used to categorize poetry into various emotion categories, including love, anger, loneliness, suicide, and surprise [52].

2.3. Automatic Classification of Poems

The computational comprehension of poetry has been studied previously.

- **English**: Hayward contended that the metrical choice of poets demonstrates distinction and personality. This is an early study of poetry classification. Hayward used a connectionist model to successfully discover substantive differences among the ten surveyed poets [5]. Subsequently, research on the classification of English poetry continued. Kaplan and Blei proposed automatically recognizing poetry style through computers. Their experiment used a quantitative method to evaluate the style of American poetry and PCA to describe the relationship between poetry collections [6]. Kao and Jurafsky compared the styles of award-winning poets and amateur poets according to quantitative characteristics and revealed the aesthetics of linguistic art using computer technology [7]. Lou et al. concentrated on how the lexicon of a poem determines its theme. They used tf-idf and latent Dirichlet allocation for the feature extraction of English poems and SVM for classification [8].
- **Persian**: Hamidi et al. used SVM to classify spoken-word Persian poems based on a syllable system [9].
- **Ottoman**: Can et al. used SVM and NB as classification algorithms to classify the poets and ages of Ottoman poetry. The results showed that SVM is a more accurate classifier than NB [10].
- **Malaysian**: Jamal et al. tried to classify Malaysian poetry by theme, poetry and nonpoetry. The results proved SVM's ability to classify poems [11].
- **Spanish**: Barros et al. used DT to classify Francisco Quivedo's poems. Manuel Blecua's classification of Francisco Quevedo's poetry's emotion was used as a reference classification. The experimental results showed that the classification model that they established could automatically classify poetry emotion with an accuracy of 56.22%. After filtering, the accuracy was increased to 75.13% [12].
- **Arabic**: Alsharif et al. built an Arabic poetry corpus with emotional annotation to examine and evaluate the impact of different levels of language preprocessing settings, feature vector dimensions, and machine learning methods on emotion classification, and studied and compared four machine learning algorithms. Hyperpipes was shown to have the highest accuracy at 79% [13].
- **Bengali**: Rakshit used semantic features to classify the themes of Bengali poetry. They used SVM to divide Tagore's poetry into four categories. They also found the most useful lexical features in each category. The accuracy of the classifier was 56.8%. The experiment also used the SVM classifier to identify the four poets, with an accuracy of 92.3% [14].
- **Punjabi**: Kaur and Saini conducted a lexical and content-based categorization of Punjabi poetry, evaluating the classification accuracy of eleven machine algorithms. According to the classification results, Hyperpipes, KNN, NB, and SVM had efficiencies ranging from 50.63% to 52.92% and 52.75% to 58.79%, respectively, surpassing the performance of all other tested machine learning algorithms [15].
- **Chinese**: Li et al. explained the feasibility of a poetry style evaluation method based on term connection and used this method to determine poetry style [16]. Yi et al. classified song lyrics into two artistic categories, the Magnificent School and the Euphemistic School, using NB. An accuracy rate of 88.5% was achieved [17]. Fang et al. studied the problem of computer-aided understanding and analysis of Chinese classical poetry and built a syntactic parser that could automatically identify

poetry images [18]. Voigt and Jurafsky utilized computer-assisted analysis for feature extraction to examine the link between new Chinese poetry and classical Chinese poetry [19].

The focus of computer-assisted poetry research has been on analyzing and studying poetry, with limited attention given to the educational aspects of the field. However, the crucial role that researchers and learners play in the development of poetry education cannot be overstated. Their actions have a direct impact on its sustainability and growth. Researchers have the potential to enrich and develop theories of poetry, while learners, armed with a deep understanding and study of poetry, can become creators of poetry, providing a wealth of material for poetry research. By continually promoting innovation in poetry education, a better research and learning environment can be created for both researchers and learners. This paper aims to redirect the attention of computer-assisted poetry research toward poetry education by building on past research on automatic poetry classification. The proposed model intends to promote sustainable and high-quality poetry education.

3. Modern Chinese Poetry Prediction Framework

In this section, the specific training process of the Chinese modern poetry prediction framework model is introduced, as shown in Figure 1. First, the data are preprocessed; then, the poem vector is generated through Doc2Vec. Furthermore, the datasets are divided into training sets and test sets, which are used to train the models. Finally, the real datasets are used for experiment and evaluation and to compare the four algorithms.

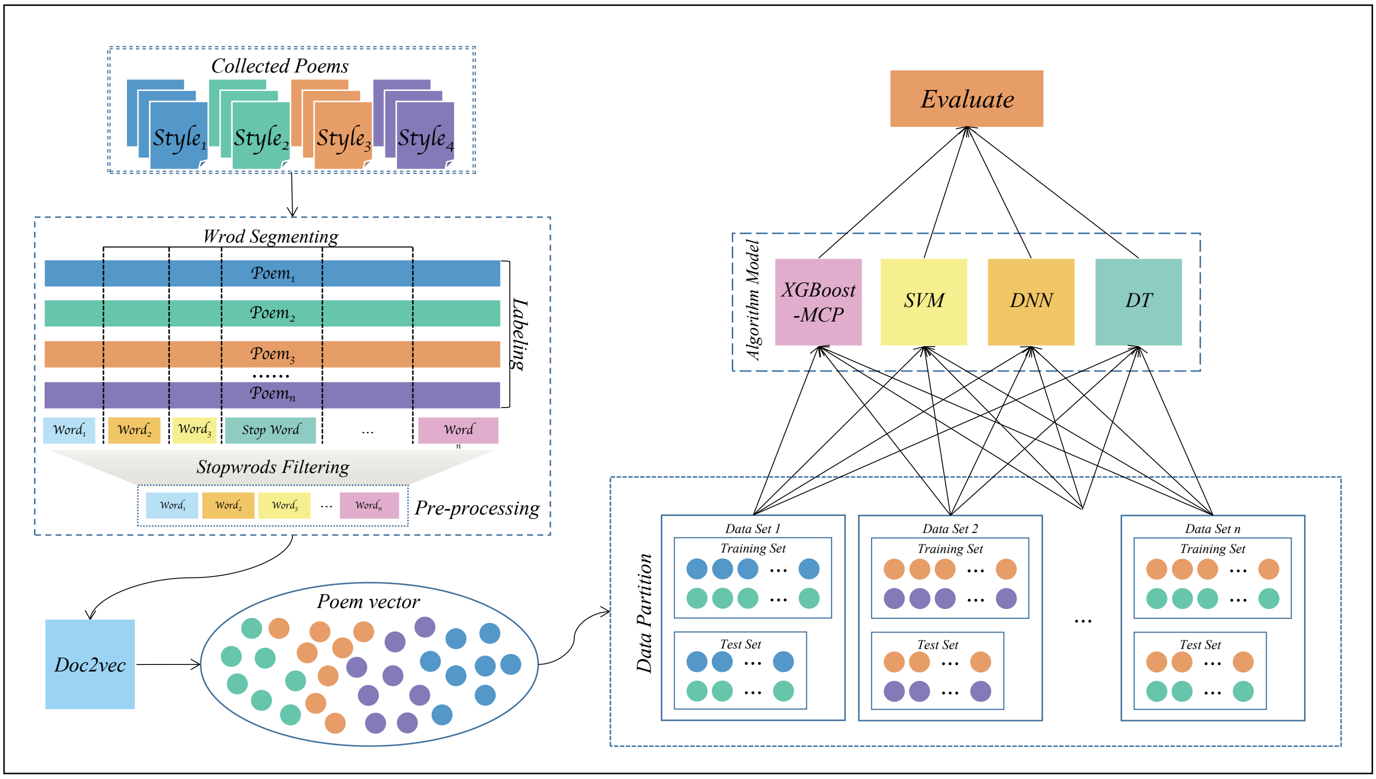

Figure 1. Chinese modern poetry prediction framework training process structure diagram.

3.1. Data Preprocessing

Text preprocessing is a key step in text classification. The results of word segmentation and stop word filtering will directly affect the results of feature extraction and thus the results of text classification. Text preprocessing is divided into three steps.

- **Labeling**: Based on the supervised learning approach, the style to which each poem belongs is manually labeled to facilitate the following classification.
- **Word segmenting**: Word segmenting is an important step in natural language processing. This is a process that regroups the sentences formed by character sequences into a set of words according to certain rules. Chinese word segmentation is much more complicated than English word segmentation. Jieba is a widely used Chinese word-splitting tool with a good word-splitting effect.
- **Stopword filtering**: The accuracy of stopword filtering directly affects the results of text analysis. Words without practical meaning, such as auxiliary words, interfere with the accuracy of the word-splitting results.

3.2. Feature Extraction

If these data are directly used as eigenvectors after text preprocessing, the dimension of the feature vector may reach tens of thousands or even hundreds of thousands. Therefore, a feature selection method is required for processing to reduce feature dimensions and improve classification efficiency. In this paper, the Doc2Vec method was used for feature extraction.

Doc2Vec is an unsupervised framework, and this method can be applied to variable-length texts, any sentence, or large documents. The process of training the poetry vector is shown in Figure 2. The input layer contains the poem vector and the word vector in the poem. Each training is completed by sliding the window to intercept part of the Chinese word-splitting vector. Therefore, the same poem will have multiple training processes, and the poem vector will share several training processes of the same poem. The poem vector can be regarded as the main idea or as representative of the poem. The theme of the poem is included in each training process as part of the input. In each training process, not only do the trained words obtain word vectors, but they also obtain the shared poem vector, which is part of the input layer of each training process. This is increasingly accurate in the training of sliding several words. When Doc2Vec is used to train the poem vector, each poem is represented by a unique vector, which is equivalent to training the poem vector and the words in the poem together to calculate the maximum likelihood of learning the relationship.

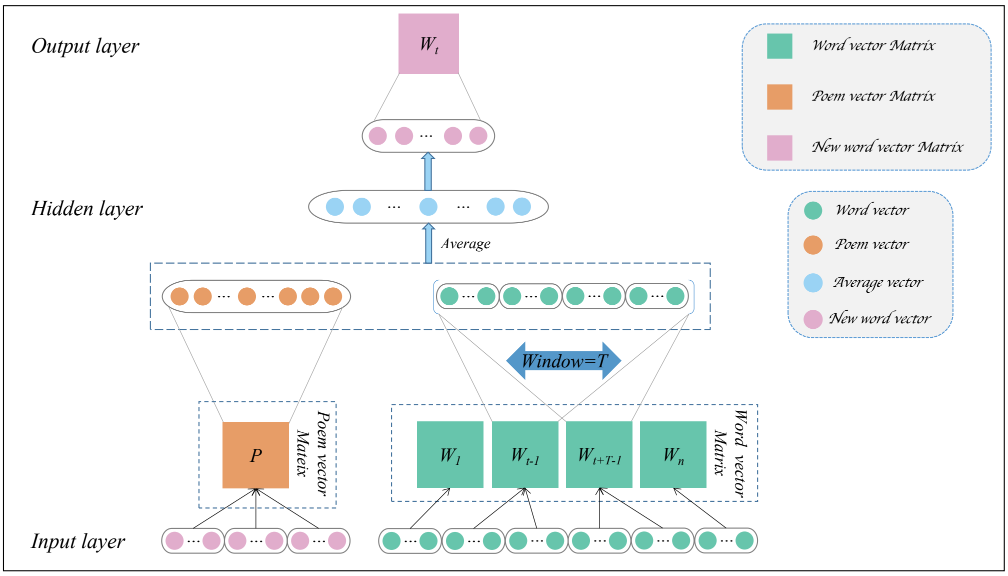

Figure 2. Poem vector training based on Doc2Vec.

The poem vector P and context word vector w are combined by taking the average or concatenation method and are expressed by the following formula

$$w(\tilde{x}) = (P, W_{t-1}, W_{t+1}, W_{t+2}, \ldots), \tag{1}$$

where $w(\tilde{x})$is the respective vector of words and W_t is the t-th word. Next, the results of the combination of poem vector P and word vector w are processed by the activation function, and then the results are added as

$$y = \mathrm{VW}(P, W_{t-1}, W_{t+1}, W_{t+2\ldots}) + b, \tag{2}$$

where V and b represent the parameters of the activation function. Assume that word vectors closer to the predicted word are labeled 1 and the those further away are labeled 0. Then, the typical sigmoid function is chosen here to calculate the probability of words being labeled 1 with the following formula

$$\sigma\left(w(\tilde{x})^T\theta^{\tilde{x}}\right) = \frac{1}{1+e^{-w(\tilde{x})^T\theta^{\tilde{x}}}}, \tag{3}$$

and the probability of words being labeled 0 is $1-\sigma\left(w(\tilde{x})^T\theta^{\tilde{x}}\right)$. $\theta^{\tilde{x}}$ refers to the vector of non-leaf nodes in a Huffman tree corresponding to the word vector in the output layer. The above two probability formulas are substituted into the function to be optimized and simplified as follows

$$L(\tilde{x}) = \left(1-\mathrm{d}^{\tilde{x}}\right)\log\left[\sigma\left(w(\tilde{x})^T\theta^{\tilde{x}}\right)\right] + \mathrm{d}^{\tilde{x}}\log\left[1-\sigma\left(w(\tilde{x})^T\theta^{\tilde{x}}\right)\right], \tag{4}$$

where $\mathrm{d}^{\tilde{x}}$ is the Huffman encoding corresponding to the word vector. Then, the random gradient descent method is used to train P and w. The objective function is optimized to obtain the optimal vector. The updated formula during window movement and model training iteration is

$$\theta^{\tilde{x}} = \theta^{\tilde{x}} - \gamma\left\{1-\mathrm{d}^{\tilde{x}} - \sigma\left(w(\tilde{x})^T\theta^{\tilde{x}}\right)\right\}w(\tilde{x}), \tag{5}$$

where γ is learning rate. In addition, the updated formula $w(\tilde{x})$ is

$$w(\tilde{x}) = w(\tilde{x}) - \gamma\sum\frac{\partial L(\tilde{x})}{\partial w(\tilde{x})}. \tag{6}$$

In the model training, the model parameters are updated from the gradient calculation error of the neural network. After convergence, the accuracy of the vector is continuously improved. The poem vector can increasingly represent the theme of the poem and can also obtain word vectors that can be processed into a format that can be provided for machine learning classification algorithms.

3.3. Introduction of the XGBoost-MCP Algorithm

The XGBoost algorithm proposed by Chen et al. is an integrated tree model, and this improved Gradient Boosting Decision Tree algorithm has the advantages of fast training and high prediction accuracy [53]. In 2022, Shwartz-Ziv and Armon compared the classification and regression task performance of the tree integration model XGBoost and several deep learning models on different tabular datasets. The results showed that XGBoost consistently outperformed deep learning models, including those used in previous papers, claiming that deep models have superior performance. They suggested that XGBoost should still be considered the first choice in data science projects [54]. Therefore, in this paper, we chose the XGBoost algorithm to build the XGBoost-MCP classification model. Table 1 of symbols in the XGBoost-MCP algorithm is as follows.

Table 1. Table of symbols in the XGBoost-MCP algorithm.

Symbol	Explanation
f	A tree structure model
T	The number of nodes
ω	The weight matrix of the T dimension
m	The number of features
q	Mapping data x from the m dimension to a node
L	The loss function
C	Constant
y_i	The true value of the i-th data point
$\hat{y}_i$	The predicted value of the i-th data point of the current tree
$\hat{y}_i^{t-1}$	The predicted value of the i-th data point of the previous tree
f_t	The output of the current tree model
$\Omega(f_t)$	The regularization term
γ	The complexity cost introduced by adding new leaf nodes T
λ	The coefficient of the regular term
$\mathrm{L}\left(\mathrm{y_i}, \hat{y}_i^{t-1}\right)$	The cumulative loss value of the prediction result of the previous tree
g_i	The first order derivative of $\mathrm{L}\left(\mathrm{y_i}, \hat{y}_i^{t-1}\right)$
h_i	The second order derivative of $\mathrm{L}\left(\mathrm{y_i}, \hat{y}_i^{t-1}\right)$
j	Leaf node
$\mathrm{L_j} = \{\mathrm{i} \mid q(\mathrm{x_i}) = j\}$	The sample set allocated to j
G_j	The sum of the first order derivatives in the set of samples of the current j
H_j	The sum of the second order derivatives in the set of samples of the current j
L	The left subtree node
R	The right subtree node
Obj	The value of the node before it is split
Obj_L	The value of the left leaf node
Obj_R	The value of the right leaf node

We define each tree model in XGBoost-MCP as follows

$$f_k(x) = \omega_{q(x)}; \omega \in R^{\top}, q : R^m \to T, \tag{7}$$

each f is regarded as a tree structure model. It turns a data point into a weight ω. ω is the weight matrix of the T dimension. T is the number of nodes. q means mapping x from the m dimension to a node. m is the number of features. Its function is to divide data x with m characteristics into a leaf model under a node with weight ω. XGBoost-MCP applies many transformations to the objective function to better calculate the optimal weight. The objective function is as follows

$$\mathrm{Obj} = \sum_{i=1}^{n} \mathrm{L}(y_i, \hat{y}_i) = \sum_{i=1}^{n} \mathrm{L}\left(y_i, \hat{y}_i^{t-1} + f_t(x_i)\right) + \Omega(f_t) + \mathrm{C}, \tag{8}$$

where L is the loss function, C is a constant, y_i is the true value of the i-th data point, $\hat{y}_i$ is the predicted value of the i-th data point of the current tree, and $\hat{y}_i^{t-1}$ is the predicted value of the i-th data point of the previous tree. f_t is the output of the current tree model, and $\Omega(f_t)$ is the regularization term. This is defined as $\Omega(f_t) = \gamma T + \frac{1}{2}\lambda\|\omega\|^2$. γ is the complexity cost introduced by adding new leaf nodes, and it controls the complexity. γT is the structure of a tree, λ is the coefficient of the regular term. The Taylor expansion of the loss function is simulated using the first and second derivatives as follows

$$L(y_i, \hat{y}_i) \approx L(y_i, \hat{y}_i) + L'(y_i, \hat{y}_i)\Delta x + \frac{1}{2}L''(y_i, \hat{y}_i)\Delta x^2, \tag{9}$$

and

$$\mathrm{Obj} \approx \sum_{\mathrm{i}=1}^{\mathrm{n}} \left[\mathrm{L}\left(\mathrm{y_i}, \hat{y}_i^{t-1}\right) + \partial_{\hat{y}_i^{t-1}} \mathrm{L}\left(\mathrm{y_i}, \hat{y}_i^{t-1}\right) \mathrm{f_t}(\mathrm{x_i}) + \frac{1}{2} \partial^2_{\hat{y}_i^{t-1}} \mathrm{L}\left(\mathrm{y_i}, \hat{y}_i^{t-1}\right) \mathrm{f_t^2}(\mathrm{x_i}) \right] + \Omega(\mathrm{f_t}) + \mathrm{C}, \tag{10}$$

where $\mathrm{L}\left(\mathrm{y_i}, \hat{y}_i^{t-1}\right)$ is the cumulative loss value of the prediction result of the previous tree. It is known in the current tree and does not participate in optimization. $\partial_{\hat{y}_i^{t-1}} \mathrm{L}\left(\mathrm{y_i}, \hat{y}_i^{t-1}\right)$ and $\partial^2_{\hat{y}_i^{t-1}} \mathrm{L}\left(\mathrm{y_i}, \hat{y}_i^{t-1}\right)$ are also known. They are represented by g_i and h_i, respectively. Then, f_t and $\Omega(\mathrm{f_t})$, which were defined previously, are substituted into the above equation, and we can obtain

$$\begin{aligned} \mathrm{Obj} &= \sum_{\mathrm{i}=1}^{\mathrm{n}} \left[g_i \mathrm{f_t}(\mathrm{x_i}) + \frac{1}{2} h_i \mathrm{f_t^2}(\mathrm{x_i}) \right] + \Omega(\mathrm{f_t}) \\ &= \sum_{\mathrm{i}=1}^{\mathrm{n}} \left[g_i \omega_{q(x)} + \frac{1}{2} h_i \omega^2_{q(x)} \right] + \gamma T + \frac{1}{2} \lambda \sum_{j=1}^{T} \omega_j^2 \\ &= \sum_{\mathrm{j}=1}^{\mathrm{T}} \left[\left(\sum_{\mathrm{i} \in \mathrm{L_j}} \mathrm{g_i} \right) \omega_{\mathrm{j}} + \frac{1}{2} \left(\sum_{\mathrm{i} \in L_{\mathrm{j}}} \mathrm{h_i} + \lambda \right) \omega_{\mathrm{j}}^2 \right] + \gamma T, \end{aligned} \tag{11}$$

where j is leaf node. $\mathrm{L_j} = \{\mathrm{i} \mid q(\mathrm{x_i}) = j\}$ is the sample set allocated to leaf node j. G_j is the sum of the first-order derivatives in the set of samples of the current leaf node j. H_j is the sum of the second-order derivatives in the set of samples of the current leaf node j. If $G_j = \sum_{i \in I_j} g_i$, $H_j = \sum_{i \in I_j} h_i$ is defined, and the above equation can be written as

$$\mathrm{Obj} = \sum_{\mathrm{j}=1}^{\mathrm{T}} \left[G_j \omega_{\mathrm{j}} + \frac{1}{2} (H_j) \omega_{\mathrm{j}}^2 \right] + \gamma T, \tag{12}$$

where G_j represents the sum of the first derivative in the sample set of the current leaf node j. H_j represents the sum of the second derivative in the sample set of the current leaf node j when the tree structure is fixed, that is, when $q(x)$ is fixed. The optimal solution of ω_j in the above equation is

$$\omega_j^* = -\frac{G_j}{H_j + \lambda} + \gamma T, \tag{13}$$

so the optimal solution ω_j^* is substituted into the original objective function to obtain the optimal objective function value under the current tree structure as follows

$$\mathrm{Obj} = -\frac{1}{2} \sum_{\mathrm{j}=1}^{\mathrm{T}} \frac{G_j^2}{H_j + \lambda} + \gamma T. \tag{14}$$

To minimize the loss of the loss function, it is assumed that the left subtree node is L and the right subtree node is R after splitting, so the expectation maximization is as follows

$$\begin{aligned} \mathrm{Gain} &= Obj - Obj_L - Obj_R \\ &= -\frac{1}{2} \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} + \gamma T - \left(-\frac{1}{2} \frac{G_L^2}{H_L + \lambda} - \frac{1}{2} \frac{G_R^2}{H_R + \lambda} + \gamma(T+1) \right) \\ &= \max \frac{1}{2} \frac{G_L^2}{H_L + \lambda} + \frac{1}{2} \frac{G_R^2}{H_R + \lambda} - \frac{1}{2} \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} - \gamma, \end{aligned} \tag{15}$$

where Obj is the value of the node before it is split, Obj_L is the value of the left leaf node, and Obj_R is the value of the right leaf node.

Based on the above formula, if the maximum value is 0, the current decision tree is established. After calculating the optimal solution of all leaf nodes, the weak learner is obtained, and then the strong learner is updated to enter the next round of weak learner iteration.

3.4. Application in Education

Figure 3 demonstrates the powerful connection between computer technology and poetry education through the application of the XGBoost-MCP model in literature education. The model offers significant advantages to learners and researchers of poetry, allowing them to classify poems accurately and objectively without the need for extensive experience. Traditionally, learners would have to spend a lot of time and effort reading to gain the necessary literary experience required for accurate poetry classification. However, the XGBoost-MCP model proposed in this paper provides learners with a faster, more efficient approach to classification that saves them time and energy.

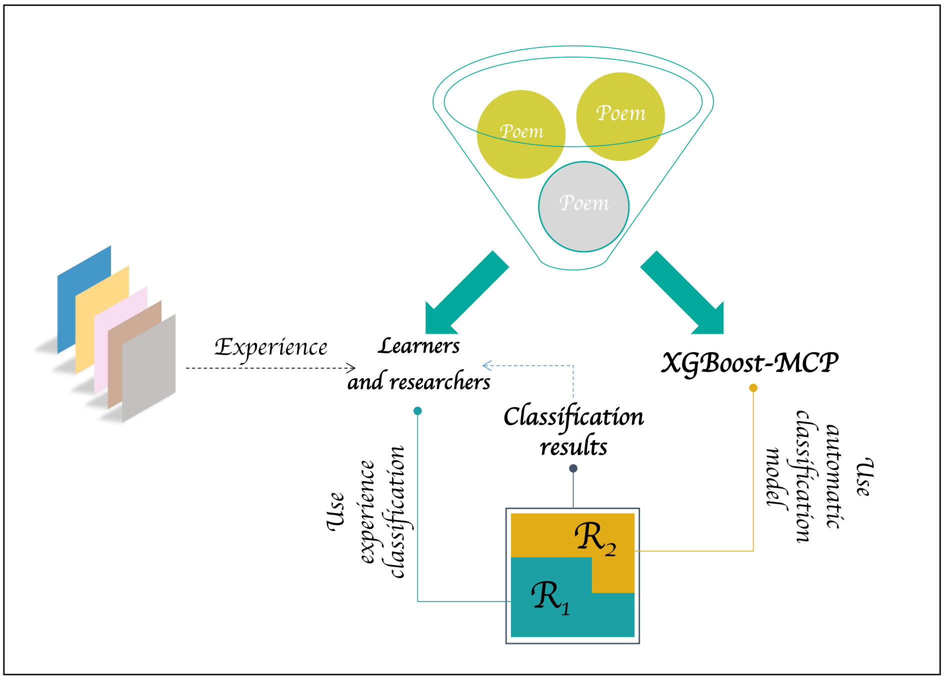

Figure 3. The application of the XGBoost-MCP model to assist poetry learners in their literary education.

Furthermore, the model overcomes the limitations of subjective experience and personal bias in poetry classification that are characteristic of traditional education in literary poetry. By providing more valuable classification references for literary researchers in the field of poetry research, the model facilitates academic research and contributes to the sustainable development of poetry learning and research. This paper presents an innovative approach to poetry education, using an artificial-intelligence-based interactive learning environment for literary poetry education. Learners and researchers are the subjects of poetry education and benefit greatly from the integration of computer technology into the field. Through this environment, they are able to access better quality poetry education and further their knowledge and understanding of poetry, leading to new insights and perspectives on this timeless art form.

4. Experiment

4.1. Datasets

Data collection is an important step when training classification models. This paper focuses on the classification of new Chinese poetry. For this reason, we collected existing collections of relevant poems and obtained a total of 836 samples. The opinions of researchers of contemporary literature were used as a reference when manually labeling four categories: Crescent School, Symbolism School, Modernism School, and Nine Leaves School. The sample data were eventually generated and divided into training and test sets at a ratio of 753:83 based on the size of the data, which was obtained using the hold-out method, as shown in Table 2.

Table 2. Distribution of the training and test sets for each poetry type in the sample data.

The Case Category	Training Sets	Test Sets
Crescent School	201	22
Symbolism School	223	25
Modernism School	183	20
Nine Leaves School	146	16

The classification models we used, including SVM and DT, are typical models used for binary classification. Although they can also be used for multiclassification, the effect is not as good as it is for binary classification. If we use multiclassification to conduct experiments, it cannot reflect the optimal performance of these models; therefore, we chose the two styles as a group to divide the dataset. The training sets and test sets shown in Table 2 were used to combine the four styles into pairs to obtain six datasets, as shown in Table 3. T001 includes the test sets and training sets of the Nine Leaves Sect and the Crescent Sect, with a size of 385. T002 includes the test sets and training sets of the Nine Leaves School and the Symbolism School, with a size of 410. T003 includes the test sets and training sets of the Modernism School and the Nine Leaves School, with a size of 365. T004 includes the test sets and training sets of the Modernism School and the Crescent School, with a size of 426. T005 includes the test sets and training sets of the Modernism School and the Symbolism School, with a size of 451. T006 includes the test sets and training sets of the Crescent School and the Symbolism School, with a size of 471.

Table 3. Datasets used in the experiment.

Datasets	The Case Category	Size
T001	Nine Leaves School and Crescent School	385
T002	Nine Leaves School and Symbolism School	410
T003	Modernism School and Nine Leaves School	365
T004	Modernism School and Crescent School	426
T005	Modernism School and Symbolism School	451
T006	Crescent School and Symbolism School	471

4.2. *Evaluation Method*

To test the performance of the algorithm, accuracy, precision, recall, and F1-scores were used as evaluation indices. These evaluation indicators were obtained based on the confusion matrix.

(1) Accuracy: The accuracy rate is the percentage of correct prediction results in the total sample. For a binary confusion matrix, the results can be classified into four categories: true positive (*TP*), false negative (*FN*), false positive (*FP*), and true negative (*TN*). The formula of the accuracy is

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN}. \tag{16}$$

(2) Precision: Precision refers to the prediction results, meaning the probability of true positive samples among all predicted positive samples. The formula of precision is

$$\text{Precision} = \frac{TP}{TP + FP}. \tag{17}$$

(3) Recall: The probability of predicted positive samples among all actual positive samples. The formula of the recall is

$$\text{Recall } = \frac{TP}{TP + FN}. \tag{18}$$

(4) F1-score: The $F1$-score can be viewed as a harmonic mean of precision and recall. When the $F1$-score is equal to 0, it indicates the worst performance of the model; conversely, when f is equal to 1, it means that the model has the best feasibility. Accuracy and recall have equal impacts on the $F1$-score. The $F1$-score is calculated as follows

$$F1 = \frac{2 \times \text{ precision } \times \text{ recall}}{\text{precision } + \text{ recall}}. \tag{19}$$

4.3. Parameter Setting of the Doc2Vec Model

The Doc2Vec model uses the following parameters: $vector\_size = 200$, $window = 15$, $min\_count = 1$, $workers = 4$, $sample = 1e - 4$, $epochs = 20$. $vector\_size$ is the dimension of the eigenvector. $window$ is the maximum distance between the current word and the predicted word in the sentence. min_count means all words with a total frequency lower than this value are ignored. $workers$ means faster training with multicore machines. $sample$ is the threshold for random downsampling when configuring high-frequency words. $epochs$ is the number of iterations on the corpus.

4.4. Training and Testing Design

Kowsari et al. mentioned that SVM is a widely used classification technology, DT classifies text quickly and accurately, and deep learning models have also obtained state-of-the-art results in natural language processing [55]. Therefore, three widely used and excellently performing text classification models, SVM, DNN, and DT, were selected for comparison experiments with XGBoost-MCP to test the performance. The specific design is as follows.

Step1. The sparse matrix space of word vectors formed from the preprocessed sample datasets were extracted using the Doc2Vec model to generate a feature vector space; this is then formed into a poem vector for each sample and input into the machine learning model for training.

Step 2. The Doc2Vec sample data are divided into training sets and test sets R a ratio of 753:83. The training sample data were trained with four machine learning models: the XGBoost-MCP algorithm, the SVM algorithm, the DNN algorithm, and the DT algorithm.

Step 3. After the parameters are adjusted, the best model for each algorithm is trained, the real data are used for testing, and the experimental results are analyzed.

4.5. Parameter Setting of the Classifier

In this work, the four classifier parameters were set as follows:

(1) XGBoost-MCP: Before running XGBoost-MCP, we must set three types of parameters: general parameters, booster parameters, and task parameters. The general parameters determine which Booster we choose to use for Boosting. The Booster parameter depends on the Booster type that is selected. Learning task parameters determine the learning strategies. In this work, the XGBoost-MCP algorithm used the built-in default parameter settings.

(2) SVM: SVM constructs an optimal classification hyperplane to form the maximum interval and controls the performance of the classifier by controlling the interval on both sides of the classification hyperplane. The parameter settings of the SVM are as follows: $C = 2$, $kernel$ = 'rbf', $gamma$ = 10, $decision\_function\_shape$ = 'ovo'. C is the penalty coefficient. This is used to control the loss function. The greater the value of C, the greater the penalty for misclassification. In this way, the accuracy of the tested training set is high, but its generalization ability is weak, which can easily lead to overfitting. The smaller the

value of C, the smaller the penalty for misclassification, and the stronger the fault tolerance, but the generalization ability is strong, which easily leads to underfitting. *kernel* is the sum function type used in the algorithm, and the kernel function is a method used to transform a nonlinear problem into a linear problem. *gamma* is the coefficient of the kernel function. $decision\_function\_shape$ is the classified decision, and *ovo* can achieve a better classification effect.

(3) DNN: DNN is a multilayer unsupervised neural network that uses the output features of the upper layer as the input of the lower layer for feature learning. After layer-by-layer feature mapping, the features of the existing spatial samples are mapped to another feature space to learn to obtain a better feature representation for the existing input. The parameter settings of the DNN are as follows: $Embedding(vocad\_size, 100, input\_length = sentence\_size)$, $Conv1D(kernel\_size = 5, activation = \text{'}relu\text{'})$, $MaxPooling1D(pool\_size = 2)$, $Dense(4, activation = \text{'}softmax\text{'})$, $optimizer = \text{'}adam\text{'}$, $loss = \text{'}categorical\_crossentropy\text{'}4$, $metrics = [\text{'}accuracy\text{'}]$. The embedding layer can be used to reduce dimensions and extract data features. In natural language processing, if a sentence sequence is one-dimensional, Conv1d is used. At this time, the convolution kernel is one-dimensional, except for the channel. MaxPooling1D calculates the maximum value in the steps dimension. However, the pooled size of each step is limited. "Dense" represents the fully connected layer, which is equivalent to adding a layer, also known as the dense layer. The optimizer is used to calculate and update the network parameters that affect model training and model output so that they can approximate or reach the optimal value, thus minimizing or maximizing the cost function. Loss is the loss function, which is the key to determining the quality of network learning. If the network structure remains unchanged, improper selection of the loss function will lead to poor model accuracy and other consequences. Metrics are key to measuring a model.

(4) DT: DT is a basic classification and regression method. In machine learning, a decision tree is a prediction model that represents a mapping relationship between object attributes and object values. The parameter settings for the DT are as follows: $max\_depth = 3$, $random\_state = 123$. max_depth is the maximum depth of the tree. After $random\_state$ is fixed, the models built each time are the same, the datasets generated are the same, and the splitting results are the same.

4.6. Analysis of the Results

In this subsection, we describe the parameter optimization and experiments that were carried out for each model. The results of the experiment are reported in detail.

Table 4 shows the accuracy results of the XGBoost-MCP algorithm compared with three classifier algorithms. The average accuracy result of the XGBoost-MCP algorithm is 0.9362, while the average accuracy results of the SVM, DNN, and DT algorithms are 0.8729, 0.8685, and 0.9047, respectively. The average accuracy of XGBoost-MCP is 7.25%, 7.80%, and 3.48%, higher, respectively, than the average accuracies of the other methods.

Table 4. Comparison of the accuracy scores of the XGBoost-MCP, DT, DNN, and SVM models.

Data Sets / Model	T001	T002	T003	T004	T005	T006	Average
XGBoost-MCP	0.9487	0.9459	0.8536	0.9375	0.9782	0.9534	0.9362
SVM	0.8974	0.8918	0.7317	0.8750	0.9347	0.9069	0.8729
DNN	0.8708	0.9130	0.8177	0.7621	0.9347	0.9130	0.8685
DT	0.8717	0.9189	0.8780	0.9166	0.9130	0.9302	0.9047

Table 5 shows the precision results of the XGBoost-MCP algorithm compared with three classifier algorithms. The average precision result of the XGBoost-MCP algorithm is 0.9409. The average precision results of the SVM, DNN, and DT algorithms are 0.8954, 0.8912 and 0.9074, respectively. The average precision of XGBoost-MCP is 5.08%, 5.58%, and 3.69% higher, respectively, than the average precisions of the other methods.

Table 5. Comparison of the precision scores of the XGBoost-MCP, DT, DNN, and SVM models.

Model \ Data Sets	T001	T002	T003	T004	T005	T006	Average
XGBoost-MCP	0.9487	0.9519	0.8700	0.9381	0.9790	0.9577	0.9409
SVM	0.8974	0.8918	0.8454	0.8864	0.9417	0.9099	0.8954
DNN	0.8627	0.9161	0.8177	0.8913	0.9464	0.9130	0.8912
DT	0.8715	0.9317	0.8700	0.9166	0.9155	0.9393	0.9074

Table 6 shows the precision results of the XGBoost-MCP algorithm compared with three classifier algorithms. The average recall result of the XGBoost-MCP algorithm is 0.9369. The average recall results of the SVM, DNN, and DT algorithms are 0.8772, 0.8903, and 0.9046, respectively. The average recall of XGBoost-MCP is 6.81%, 5.23%, and 3.57% higher, respectively, than the average recall of the other methods.

Table 6. Comparison of the recall scores of the XGBoost-MCP, DT, DNN, and SVM models.

Model \ Data Sets	T001	T002	T003	T004	T005	T006	Average
XGBoost-MCP	0.9487	0.9459	0.8574	0.9375	0.9782	0.9534	0.9369
SVM	0.8974	0.8918	0.7575	0.8750	0.9347	0.9069	0.8772
DNN	0.8627	0.9161	0.8177	0.9007	0.9285	0.9162	0.8903
DT	0.8717	0.9189	0.8775	0.9166	0.9130	0.9302	0.9046

Table 7 shows the precision results of the XGBoost-MCP algorithm compared with three classifier algorithms. The average F1-score result of the XGBoost-MCP algorithm is 0.9365. The average F1-score results of the SVM, DNN, and DT algorithms are 0.8731, 0.8886 and 0.9047, respectively. The average F1-score of XGBoost-MCP is 7.26%, 5.39%, and 3.51% higher, respectively, than the average F1-score of the other methods.

Table 7. Comparison of the F1-scores of the XGBoost-MCP, DT, DNN, and SVM models.

Model \ Data Sets	T001	T002	T003	T004	T005	T006	Average
XGBoost-MCP	0.9487	0.9461	0.8549	0.9374	0.9782	0.9535	0.9365
SVM	0.8974	0.8922	0.7339	0.8745	0.9341	0.9064	0.8731
DNN	0.8661	0.9128	0.8177	0.8887	0.9332	0.9129	0.8886
DT	0.8710	0.9192	0.8786	0.9166	0.9125	0.9302	0.9047

Figure 4 shows the experimental data more intuitively. In the T001, T002, T004, T005, and T006 datasets, XGBoost-MCP is better than the other three algorithms in terms of accuracy precision, recall, and F1-score. As Shwartz-Ziv and Armon mentioned in their article, the XGBoost algorithm can still be used as the first-choice classification model, as also proved by its achieving the best performance in the experiment [54]. In the T003 dataset, the performance of the four classification algorithms was not as good as that of the other five datasets, which may be due to the similar style of the poetry datasets. Although the performance of XGBoost-MCP in T003 is not the best, it is not very different from the best DT obtained in the experimental results.

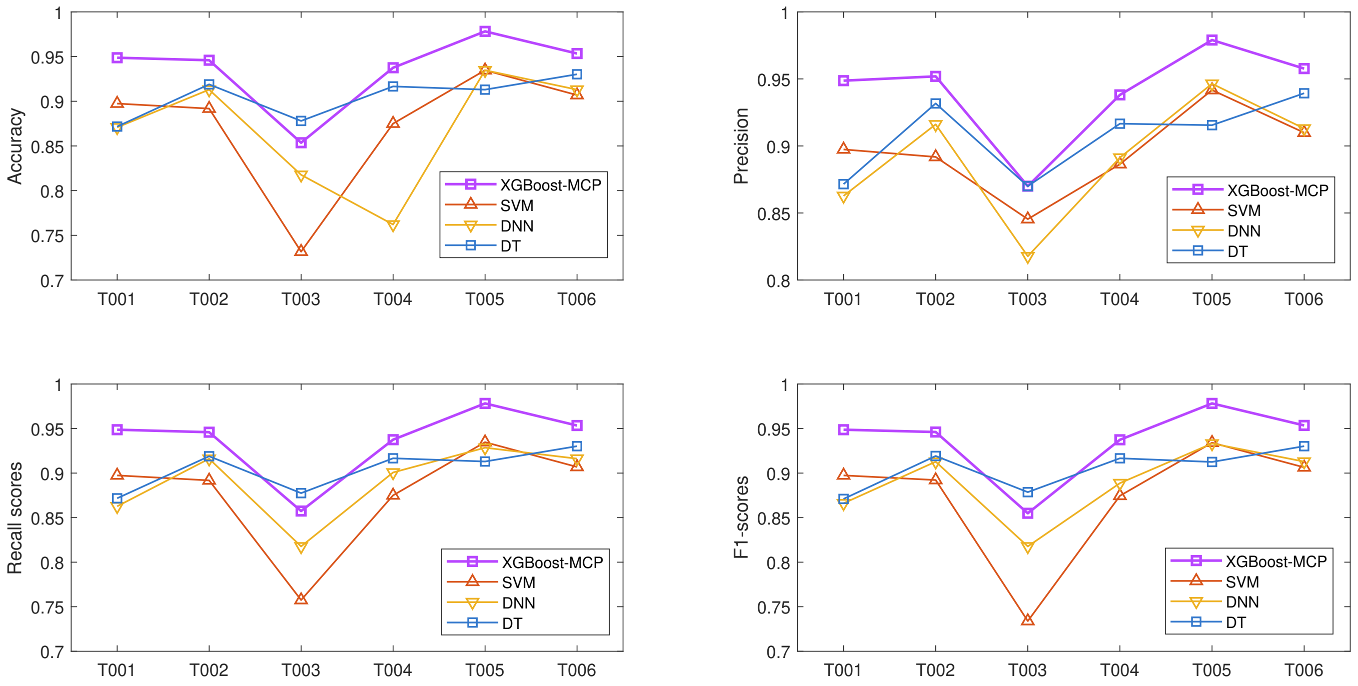

Figure 4. Comparison of experimental data of four classification models on real datasets.

Figure 5 shows that the XGBoost-MCP training model has the best performance and obvious advantages compared with the other three machine learning models. The DT training model has the second-best performance, which is superior to SVM and DNN. SVM is superior to DNN in terms of accuracy and precision, but in terms of recall and F1-score, the DNN is better.

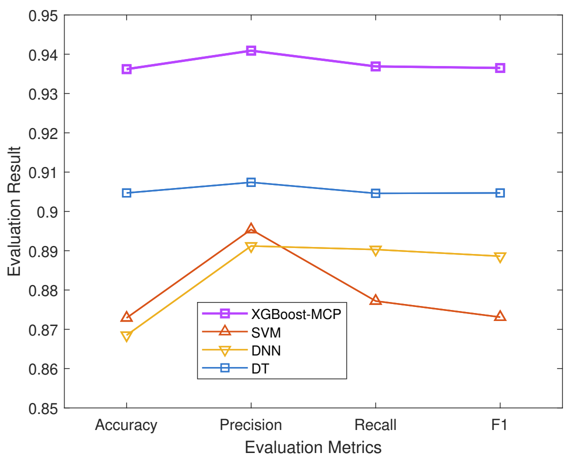

Figure 5. Comparison of experimental data, with average values for four classification models.

5. Conclusions and Prospects

Cultural heritage is an essential component of sustainable development, and education is a means of achieving it. Education is an important social activity for intergenerational communication. This paper elaborates on the use of modern artificial intelligence methods to promote modern poetry education, which is a crucial means of cultural heritage preservation and an indispensable part of sustainable development. It represents a new milestone in cultural sustainability.

After data preprocessing, we used Doc2Vec and XGBoost to obtain an XGBoost-MCP model for the automatic classification of modern Chinese poetry. The integration of computer technology into poetry education is exemplified in the development of the

XGBoost-MCP model, which offers a more accurate and efficient approach to the automatic classification of modern Chinese poetry. This model addresses the shortcomings of traditional poetry education and can provide better-quality poetry education. The application of machine learning algorithms in poetry education is an effective attempt to promote sustainable poetry education and solve the problem of classifying modern Chinese poetry.

From the perspective of the data-classification effect, we can mainly conclude that the XGBoost-MCP model showed a good performance in the automatic classification of modern Chinese poetry styles. This is obviously superior to the other three mainstream classification algorithms, which is consistent with the conclusions of Shwartz-Ziv and Armon [54]. However, only four models were selected for the analysis of experimental results; therefore, an exploration of the performance of other models in terms pf the classification of modern Chinese poetry styles should be a subject of future research. There are still a large number of poems with unspecified styles, so this classifier can be used to assist in classification. This highlights the potential of machine learning algorithms to innovate the way poetry is taught and researched. Continued research and exploration of the application of natural language processing to literary classification could lead to further improvements in classification models and contribute to the continued development of poetry education.

Author Contributions: Conceptualization, M.Z., C.L. and H.W.; methodology, M.Z., G.W., C.L. and H.W.; software, G.W. and H.W.; validation, M.Z., G.W., C.L., H.W. and B.Z.; formal analysis, M.Z., H.W. and B.Z.; Investigation, M.Z., C.L. and H.W.; Resources, C.L., H.W. and B.Z.; data curation, M.Z., G.W. and C.L.; writing—original draft preparation, M.Z. and G.W.; writing—review and editing, M.Z. and G.W.; visualization, M.Z.; supervision, C.L., H.W. and B.Z.; project administration, H.W. and B.Z.; funding acquisition, H.W. and B.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under Grant No. 62276216. And the APC was funded by Hongjun Wang.

Data Availability Statement: Our research data is published at https://github.com/mini29474081/Modern-Chinese-Poetry-Artificial-Intelligence-Classification-Model.git.

Acknowledgments: This work was funded by the National Natural Science Foundation of China under Grant No. 62276216.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had played a role in the design of the study; in the collection, analyses and interpretation of data; in the writing of the manuscript; and in the decision to publish the results.

References

1. Christou, D.; Tsoumakas, G. Extracting semantic relationships in Greek literary texts. *Sustainability* **2021**, *13*, 9391. [CrossRef]
2. Zeng, Z.; Cai, Y.; Wang, F.L.; Xie, H.; Chen, J. Weighted N-grams CNN for Text Classification. In Proceedings of the Information Retrieval Technology: 15th Asia Information Retrieval Societies Conference, AIRS 2019, Hong Kong, China, 7–9 November 2019; Springer: Cham, Switzerland, 2020; pp. 158–169.
3. Wang, T.; Cai, Y.; Leung, H.f.; Lau, R.Y.; Xie, H.; Li, Q. On entropy-based term weighting schemes for text categorization. *Knowl. Inf. Syst.* **2021**, *63*, 2313–2346. [CrossRef]
4. Wang, J.; Xie, H.; Wang, F.L.; Lee, L.K. Improving text classification via a soft dynamical label strategy. *Int. J. Mach. Learn. Cybern.* **2023**, 1–11. [CrossRef]
5. Hayward, M. Analysis of a corpus of poetry by a connectionist model of poetic meter. *Poetics* **1996**, *24*, 1–11. [CrossRef]
6. Kaplan, D.M.; Blei, D.M. A computational approach to style in American poetry. In Proceedings of the Seventh IEEE International Conference on Data Mining (ICDM 2007), Omaha, NE, USA, 28–31 October 2007; pp. 553–558.
7. Kao, J.; Jurafsky, D. A computational analysis of style, affect, and imagery in contemporary poetry. In Proceedings of the NAACL-HLT 2012 Workshop on Computational Linguistics for Literature, Montréal, QC, Canada, 8 June 2012; pp. 8–17.
8. Lou, A.; Inkpen, D.; Tanasescu, C. Multilabel subject-based classification of poetry. In Proceedings of the Twenty-Eighth International Flairs Conference, Hollywood, FL, USA, 18–20 May 2015.
9. Hamidi, S.; Razzazi, F.; Ghaemmaghami, M.P. Automatic meter classification in Persian poetries using support vector machines. In Proceedings of the 2009 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), IEEE, Ajman, United Arab Emirates, 14–17 December 2009; pp. 563–567.

10. Can, E.F.; Can, F.; Duygulu, P.; Kalpakli, M. Automatic categorization of ottoman literary texts by poet and time period. In *Computer and Information Sciences II*; Springer: London, UK, 2011; pp. 51–57.
11. Jamal, N.; Mohd, M.; Noah, S.A. Poetry classification using support vector machines. *J. Comput. Sci.* **2012**, *8*, 1441.
12. Barros, L.; Rodriguez, P.; Ortigosa, A. Automatic Classification of Literature Pieces by Emotion Detection: A Study on Quevedo's Poetry. In Proceedings of the 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction, IEEE, Geneva, Switzerland, 2–5 September 2013; pp. 141–146.
13. Alsharif, O.; Alshamaa, D.; Ghneim, N. Emotion classification in Arabic poetry using machine learning. *Int. J. Comput. Appl.* **2013**, *65*, 16.
14. Rakshit, G.; Ghosh, A.; Bhattacharyya, P.; Haffari, G. Automated analysis of bangla poetry for classification and poet identification. In Proceedings of the 12th International Conference on Natural Language Processing, Trivandrum, India, 11–14 December 2015; pp. 247–253.
15. Kaur, J.; Saini, J.R. Punjabi poetry classification: The test of 10 machine learning algorithms. In Proceedings of the 9th International Conference on Machine Learning and Computing, Singapore, 24–26 February 2017; pp. 1–5.
16. Li, L.Y.; He, Z.S.; Yi, Y. Poetry stylistic analysis technique based on term connections. In Proceedings of the 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 04EX826), IEEE, Shanghai, China, 26–29 August 2004; Volume 5, pp. 2713–2718.
17. Yong, Y.I.; Zhong-Shi, H.E.; Liang-Yan, L.I.; Zhou, J.Y. A Traditional Chinese Poetry Style Identification Calculation Improvement Model. *Comput. Sci.* **2005**, *32*, 156–158.
18. Fang, A.C.; Lo, F.J.; Chinn, C.K. Adapting nlp and corpus analysis techniques to structured imagery analysis in classical chinese poetry. In Proceedings of the Workshop on Adaptation of Language Resources and Technology to New Domains, Borovets, Bulgaria, 17 September 2009; pp. 27–34.
19. Voigt, R.; Jurafsky, D. Tradition and modernity in 20th century Chinese poetry. In Proceedings of the Workshop on Computational Linguistics for Literature, Atlanta, GA, USA, June 2013; pp. 17–22.
20. Chen, X.; Xie, H.; Hwang, G.J. A multi-perspective study on artificial intelligence in education: Grants, conferences, journals, software tools, institutions, and researchers. *Comput. Educ. Artif. Intell.* **2020**, *1*, 100005. [CrossRef]
21. Hwang, G.J.; Xie, H.; Wah, B.W.; Gašević, D. Vision, challenges, roles and research issues of Artificial Intelligence in Education. *Comput. Educ. Artif. Intell.* **2020**, *1*, 100001. [CrossRef]
22. Hwang, G.J.; Tu, Y.F. Roles and research trends of artificial intelligence in mathematics education: A bibliometric mapping analysis and systematic review. *Mathematics* **2021**, *9*, 584. [CrossRef]
23. Chang, I.C.; Yu, T.K.; Chang, Y.J.; Yu, T.Y. Applying text mining, clustering analysis, and latent dirichlet allocation techniques for topic classification of environmental education journals. *Sustainability* **2021**, *13*, 10856. [CrossRef]
24. Renfen, H.; Yuchen, Z. Automatic classification of tang poetry themes. *Acta Sci. Nat. Univ. Pekin.* **2015**, *2*, 262–268.
25. Ahmed, M.A.; Hasan, R.A.; Ali, A.H.; Mohammed, M.A. The classification of the modern arabic poetry using machine learning. *TELKOMNIKA (Telecommun. Comput. Electron. Control)* **2019**, *17*, 2667–2674. [CrossRef]
26. Abdulfattah, O. Identifying themes in fiction: A centroid-based lexical clustering approach. *J. Lang. Linguist. Stud.* **2021**, *17*, 580–594.
27. Promrit, N.; Waijanya, S. Convolutional neural networks for thai poem classification. In *International Symposium on Neural Networks*; Springer: Cham, Switzerland, 2017; pp. 449–456.
28. Ahmad, S.; Asghar, M.Z.; Alotaibi, F.M.; Khan, S. Classification of Poetry Text Into the Emotional States Using Deep Learning Technique. *IEEE Access* **2020**, *8*, 73865–73878. [CrossRef]
29. Li, X.; Pang, J.; Mo, B.; Rao, Y.; Wang, F.L. Deep neural network for short-text sentiment classification. In *Database Systems for Advanced Applications, Proceedings of the DASFAA 2016 International Workshops: BDMS, BDQM, MoI, and SeCoP, Dallas, TX, USA, 16–19 April 2016*; Springer: Cham, Switzerland, 2016; pp. 168–175.
30. Rao, Y.; Xie, H.; Li, J.; Jin, F.; Wang, F.L.; Li, Q. Social emotion classification of short text via topic-level maximum entropy model. *Inf. Manag.* **2016**, *53*, 978–986. [CrossRef]
31. Zheng, W.; Xu, Z.; Rao, Y.; Xie, H.; Wang, F.L.; Kwan, R. Sentiment classification of short text using sentimental context. In Proceedings of the 2017 International Conference on Behavioral, Economic, Socio-Cultural Computing (BESC), IEEE, Krakow, Poland, 16–18 October 2017; pp. 1–6.
32. Liang, W.; Xie, H.; Rao, Y.; Lau, R.Y.; Wang, F.L. Universal affective model for readers' emotion classification over short texts. *Expert Syst. Appl.* **2018**, *114*, 322–333. [CrossRef]
33. Chen, X.; Rao, Y.; Xie, H.; Wang, F.L.; Zhao, Y.; Yin, J. Sentiment classification using negative and intensive sentiment supplement information. *Data Sci. Eng.* **2019**, *4*, 109–118. [CrossRef]
34. Mosteller, F.; Wallace, D.L. Inference in an authorship problem: A comparative study of discrimination methods applied to the authorship of the disputed Federalist Papers. *J. Am. Stat. Assoc.* **1963**, *58*, 275–309.
35. Holmes, D.I. Authorship attribution. *Comput. Humanit.* **1994**, *28*, 87–106. [CrossRef]
36. Forsyth, R.S.; Holmes, D.I. Feature-finding for text classification. *Lit. Linguist. Comput.* **1996**, *11*, 163–174. [CrossRef]
37. Argamon, S.; Whitelaw, C.; Chase, P.; Hota, S.R.; Garg, N.; Levitan, S. Stylistic text classification using functional lexical features. *J. Am. Soc. Inf. Sci. Technol.* **2007**, *58*, 802–822. [CrossRef]

38. Koppel, M.; Argamon, S.; Shimoni, A.R. Automatically categorizing written texts by author gender. *Lit. Linguist. Comput.* **2002**, *17*, 401–412. [CrossRef]
39. Argamon, S.; Koppel, M.; Fine, J.; Shimoni, A.R. Gender, genre, and writing style in formal written texts. *Text Talk* **2003**, *23*, 321–346. [CrossRef]
40. Walkowiak, T.; Piasecki, M. Stylometry analysis of literary texts in Polish. In *International Conference on Artificial Intelligence and Soft Computing*; Springer: Cham, Switzerland, 2018; pp. 777–787.
41. Burrows, J.F. Word-Patterns and Story-Shapes: The Statistical Analysis of Narrative Style. *Lit Linguist. Comput.* **1987**, *2*, 61–70. [CrossRef]
42. Ledger, G.; Merriam, T. Shakespeare, Fletcher, and the Two Noble Kinsmen. *Lit Linguist. Comput.* **1994**, *9*, 235–248. [CrossRef]
43. Craig, H. Authorial attribution and computational stylistics: If you can tell authors apart, have you learned anything about them? *Lit. Linguist. Comput.* **1999**, *14*, 103–113. [CrossRef]
44. Diederich, J.; Kindermann, J.; Leopold, E.; Paass, G. Authorship Attribution with Support Vector Machines. *Appl. Intell.* **2003**, *19*, 109–123. [CrossRef]
45. De Vel, O.; Anderson, A.; Corney, M.; Mohay, G. Mining e-mail content for author identification forensics. *ACM Sigmod Rec.* **2001**, *30*, 55–64. [CrossRef]
46. Zheng, R.; Li, J.; Chen, H.; Huang, Z. A framework for authorship identification of online messages: Writing-style features and classification techniques. *J. Am. Soc. Inf. Sci. Technol.* **2006**, *57*, 378–393. [CrossRef]
47. Yu, B. An evaluation of text classification methods for literary study. *Lit. Linguist. Comput.* **2008**, *23*, 327–343. [CrossRef]
48. Mu, Y. Using keyword features to automatically classify genre of Song Ci poem. In Proceedings of the Workshop on Chinese Lexical Semantics, Leshan, China, 18–20 May 2015; Springer: Cham, Switzerland, 2015; pp. 478–485.
49. Pal, K.; Patel, B.V. Automatic multiclass document classification of hindi poems using machine learning techniques. In Proceedings of the 2020 International Conference for Emerging Technology (INCET), IEEE, Belgaum, India, 5–7 June 2020; pp. 1–5.
50. Kalcheva, N.; Karova, M.; Penev, I. Comparison of the accuracy and the execution time of classification algorithms for Bulgarian literary works. In Proceedings of the 2020 International Conference Automatics and Informatics (ICAI), IEEE, Varna, Bulgaria, 1–3 October 2020; pp. 1–5.
51. Wei, S.; Hou, H.; Sun, H.; Li, W.; Song, W. The Classification System of Literary Works Based on K-Means Clustering. *J. Interconnect. Netw.* **2022**, *22*, 2141001. [CrossRef]
52. Khattak, A.; Asghar, M.Z.; Khalid, H.A.; Ahmad, H. Emotion classification in poetry text using deep neural network. *Multimed. Tools Appl.* **2022**, *81*, 26223–26244. [CrossRef]
53. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM Sigkdd International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794.
54. Shwartz-Ziv, R.; Armon, A. Tabular data: Deep learning is not all you need. *Inf. Fusion* **2022**, *81*, 84–90. [CrossRef]
55. Kowsari, K.; Jafari Meimandi, K.; Heidarysafa, M.; Mendu, S.; Barnes, L.; Brown, D. Text classification algorithms: A survey. *Information* **2019**, *10*, 150. [CrossRef]