**Mathematical Modeling and Algorithm Application**
ISSN: 3006-0842 | Vol. 6, No. 1, 2025

# Research on English Film Review Classification Based on NLP

**Jiangyang Xu[*,#], Xicheng Yang[#], Zijin Li[#]**

School of Computer Science and Information Engineering, Changzhou Institute of Technology, Changzhou 213032, China

[*] **Corresponding author**: Jiangyang Xu (Email: naasi0306@outlook.com)

[#]**These authors contributed equally.**

**Abstract:** This study is based on the IMDB film review dataset released by Stanford University, and constructs a film review analysis model that integrates traditional machine learning and deep learning, with statistical methods as the main approach, to achieve sentiment classification and theme modeling of English film reviews. In the emotion classification section, this article focuses on using statistical feature extraction and modeling methods. Firstly, spaCy is used to perform preprocessing such as word segmentation and form restoration. Then, the TF-IDF algorithm is used to extract text features, and a statistical classification model based on logistic regression is constructed. After grid search optimization, the model achieves an accuracy of 88.0% on the test set. Meanwhile, by combining Word2Vec word vectors with a bidirectional LSTM neural network to design a deep learning path, with the support of regularization mechanisms such as dropout, the accuracy was improved to 88.24%. Finally, based on the validation set, a fusion strategy was proposed to combine the logistic regression and LSTM model prediction results with a weighted fusion of 0.45: 0.55, further improving the accuracy to 90.2%. In terms of topic modeling, the study adopts the classic LDA model to divide comment texts into 15 topic categories (such as post viewing sentiment, comedy, social issues, etc.), and analyzes the topic distribution of each comment through D-T matrix to achieve content dimension understanding beyond emotions.

**Keywords:** English Film Reviews, Deep Learning, Convolutional Neural Networks, Long Short-term Memory Networks, Word2Vec.

## 1. Introduction

With the rapid development of deep learning technology, neural network-based data text classification methods have achieved significant results[1]. Compared to traditional dependency rules and statistical methods, deep learning methods can automatically extract features from massive data, demonstrating higher classification accuracy.

How to organically integrate deep learning with traditional methods and build efficient and accurate data text classification models has become one of the core issues in current research. The aim of this study is to help film distributors and marketers gain a deeper understanding of audience preferences, emotional reactions, and focal points by implementing high-precision sentiment classification and topic modeling. This will provide accurate model support for English film promotion strategies and marketing. The research on data text classification in China has undergone an evolution from rule-based to statistical learning and then to deep learning. In the early days, keyword matching and rule building methods were mainly used, but these methods were difficult to handle complex texts. With the development of machine learning, Word Frequency Inverse Document Frequency (TF-IDF) and Bag of Words (BoW) models have become mainstream, and models such as Support Vector Machine (SVM) and Naive Bayes have been widely used, especially in Chinese text classification, achieving good results[2]. In recent years, the introduction of deep learning technology has led to breakthroughs in Chinese text classification methods. Based on word embedding models (Word2Vec) and deep learning models such as convolutional neural networks (CNN) and long short-term memory networks (LSTM), semantic features of text can be automatically learned, significantly improving classification accuracy[3]. Meanwhile, ensemble learning methods have been widely applied in China, enhancing the robustness of models. Pre trained models based on Transformer have shown outstanding performance in English sentiment analysis and comment classification. The research on data text classification in foreign countries started earlier, initially relying mainly on rule-based and traditional machine learning methods such as TF-IDF and SVM. On this basis, researchers have proposed more optimization algorithms, such as random forests and gradient boosting trees, to improve classification performance.

With the rise of deep learning, models such as Word2Vec, CNN, and LSTM have become mainstream, especially Transformer based pre trained language models such as BERT, which significantly improve the effectiveness of text classification, especially in tasks such as sentiment analysis and topic modeling. Foreign research continues to drive large-scale datasets and cross disciplinary applications, pushing the boundaries of text classification. This study focuses on sentiment classification and topic modeling, integrating traditional machine learning and deep learning methods, and proposing innovative model fusion strategies to enhance the reliability of film review data analysis.

## 2. Research on English Film Critics Based on TF-IDF Algorithm

### 2.1. Traditional TF-IDF feature extraction

In this study, we used the TF-IDF algorithm to convert text into numerical vectors and extract key information from the text from a statistical perspective.

TF-IDF is a commonly used text feature extraction method,

widely used in fields such as information retrieval, text classification, and sentiment analysis. The core idea is to measure the relationship between the frequency of a word appearing in a document and its wide distribution throughout the corpus.

TF-IDF consists of two parts: Word Frequency (TF) and Inverse Document Frequency (IDF).

TF represents the frequency of occurrence of a word in a document, calculated as the ratio of the number of times a word appears in the document to the total number of words in the document, reflecting the importance of the word in a single document.

$$TF(t,d) = \frac{n_{t,d}}{\sum_{w \in d} n_{w,d}} \quad (1)$$

IDF measures the prevalence of a word in the entire corpus, expressed as the logarithm of the ratio of the total number of documents in the corpus to the number of documents containing the word. The larger the IDF, the less common the word is in the corpus and the more meaningful it is for distinguishing different documents.

$$IDF(t,D) = ln\frac{|D|}{|\{d \in D | t \in d\}|} \quad (2)$$

The TF-IDF value is the product of TF and IDF. The larger its value, the more frequently the word appears in the document and has strong discriminability, making it more important in feature extraction. Through TF-IDF, representative keywords can be effectively extracted from text for text classification and analysis.

$$TF\text{-}IDF(t,d,D) = TF(t,d) \times IDF(t,D) \quad (3)$$

## 2.2. Deep Learning Word2Vec Feature Extraction

Word2Vec is a model used to map words into vector representations, belonging to a word embedding technique in the field of deep learning. Its goal is to convert words into points in a vector space that can be understood by machines through mathematical methods. In this way, semantically similar words are clustered together in the vector space to improve the effectiveness of natural language processing tasks.

The advantages of Word2Vec are high efficiency, the ability to process large-scale corpus data, and the ability to generate low dimensional word vectors, reducing computational complexity. The data flow diagram of Word2Vec is shown in Figure 1.

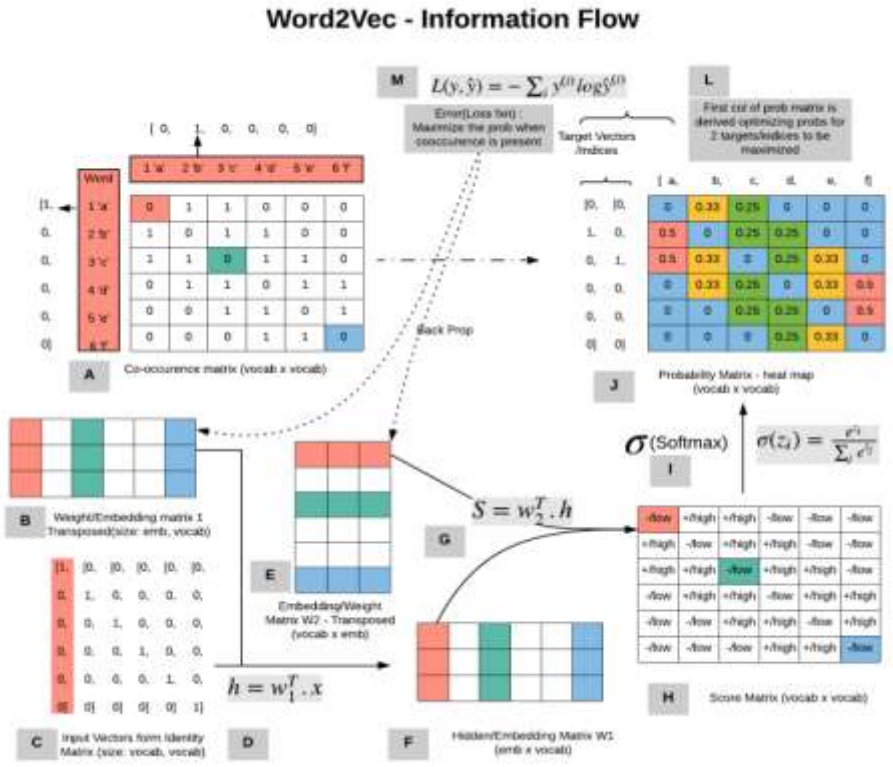


**Figure 1** Word2Vec Data Flow Diagram

In this study, word vectors were obtained through Word2Vec training, which not only preserves the grammatical information of words but also reflects certain semantic information. This has made it widely used in tasks such as text classification, sentiment analysis, and machine translation.

## 2.3. Establishment of Data Classification Model

The data classification section mainly revolves around classification models based on logistic regression. Logistic regression is a classic and widely used statistical learning method for binary classification problems, which has the advantages of solid theoretical foundation and easy implementation.

The logistic regression method introduces a non-linear mapping function (sigmoid function) by weighting and summing the input features, mapping the linear combination result to a probability value between 0 and 1, representing the likelihood of an event occurring.

### 2.3.1. Logistic regression model

Assuming there is an input feature vector $\mathbf{x} = [x_1, x_2, \ldots, x_n]$ in this article, the goal is to predict the

probability $P(y=1|\mathbf{x})$ of the event being class 1. The logistic regression assumption is that the probability of category 1 can be modeled using the sigmoid function:

$$P(y=1|\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x}+b) \quad (4)$$

Among them, σ (z) is the sigmoid function, defined as:

$$\sigma(z) = \frac{1}{1+e^{-z}} \quad (5)$$

It is the weight vector of the model and the bias term. The output of the sigmoid function is the probability that the input feature belongs to class 1. The sigmoid function graph is shown in Figure 2.

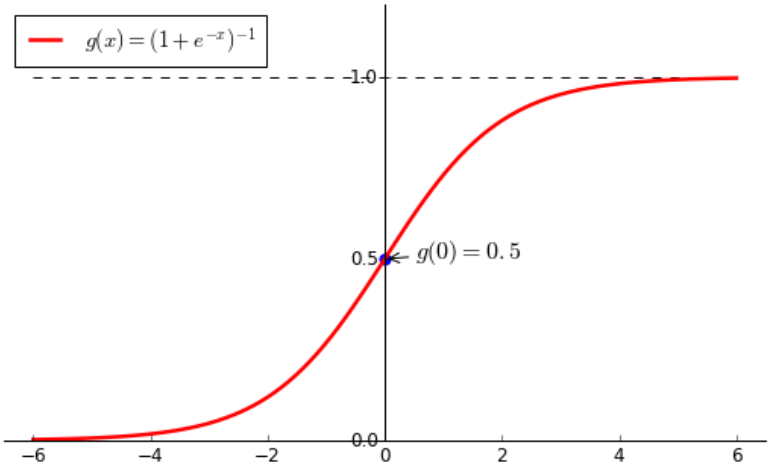


**Figure 2** Sigmoid function image

#### 2.3.2. Loss function

To train the logistic regression model, we defined a loss function to measure the difference between the probability predicted by the model and the true label[4].

The Cross Entropy Loss Function is a commonly used loss function in classification problems, which measures the distance between the predicted probability and the true class label.

For a single sample, the cross entropy loss function is defined as:

$$L(y,\hat{y}) = -[ylog(\hat{y}) + (1-y)log(1-\hat{y})] \quad (6)$$

Among them, y is the true label, and $\hat{y} = P(y=1|\mathbf{x})$ is the predicted probability. For the entire training set, the total cross entropy loss is the sum of the losses of all samples, and the average is usually taken to represent the overall loss of the model:

$$J(\mathbf{w},b) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\log(\hat{y}^{(i)}) + (1-y^{(i)})\log(1-\hat{y}^{(i)})] \quad (7)$$

Among them, m is the total number of samples, $y^{(i)}$ is the true label of the th sample, and $\hat{y}^{(i)}$ is the predicted probability of the th sample[5].

## 2.4. Model training method

#### 2.4.1. Gradient descent method

The goal of training a logistic regression model is to minimize the cross entropy loss function[6]. The common optimization algorithm is gradient descent, which iteratively updates the weight w and bias b to gradually reduce the value of the loss function, thereby obtaining an optimal model.

The update formula for gradient descent is:

$$\mathbf{w} := \mathbf{w} - \eta\frac{\partial J(\mathbf{w},b)}{\partial \mathbf{w}} \quad (8)$$

$$b := b - \eta\frac{\partial J(\mathbf{w},b)}{\partial b} \quad (9)$$

Among them, $J(\mathbf{w},b)$ is the cross entropy loss function, $\mathbf{w}$ is the weight vector of the model, $b$ is the bias term of the model, $\eta$ is the learning rate used to control the step size of each update, and the gradient descent diagram is shown in Figure 3.

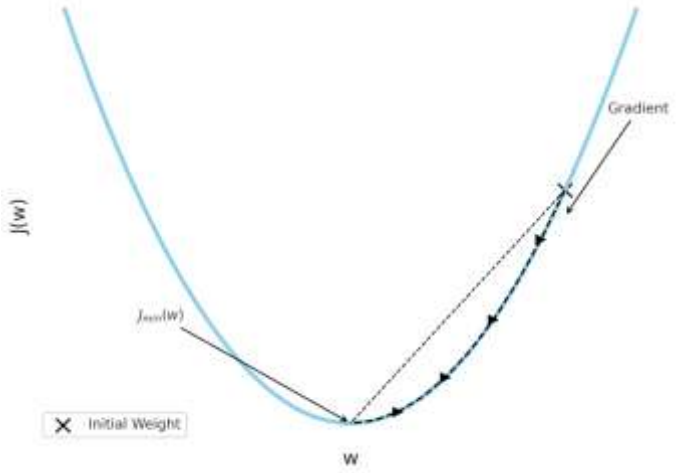


**Figure 3** Gradient Descent Diagram

#### 2.4.2. Optimizer selection

To improve the efficiency of parameter updates and convergence speed, this article chooses the Adam optimizer. Adam combines the advantages of momentum method and RMSprop, effectively dealing with sparse gradient and noisy data by adaptively adjusting the learning rate of each parameter. Adam's update rule can be expressed as:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} \hat{m}_t \tag{10}$$

Among them, $\theta_t$ represents the current parameter, $\eta$ is the preset learning rate, $\hat{m}_t$ and $\hat{v}_t$ are the first moment (mean) and second moment (variance) estimates after deviation correction, respectively, and $\epsilon$ is the numerical stability constant to prevent zero division errors.

This update strategy enables the model to utilize historical gradient information and respond to current gradient fluctuations at each adjustment step, ensuring a stable and efficient training process.

## 2.5. Model configuration and training strategy

#### 2.5.1. Training parameter settings

Setting the maximum iteration count to max_iter=1000 provides sufficient training iterations for the model. This approach provides the model with more opportunities to adjust parameters in a complex high-dimensional feature space, thereby enabling deeper learning of the implicit information and emotional patterns in textual data.

In the actual training process, the setting of batch size has a significant impact on the convergence speed and generalization performance of the model. We set batch size to 32, which ensures that each batch contains sufficient information and introduces a certain degree of randomness, making gradient estimation smoother and thus improving the model's generalization ability.

Set the number of training epochs to 10 to ensure that the model fully learns data features within a limited time. After each round of training, real-time monitoring of loss and accuracy changes through the validation set can help identify potential overfitting issues or situations where learning rate settings are unreasonable in advance.

#### 2.5.2. Data partitioning

In terms of data partitioning, this article uses the configuration of validation_stplit=0.2 to use 20% of the training data as the validation set. This partitioning method not only ensures sufficient data for model parameter updates, but also provides an independent validation set for real-time evaluation, timely feedback on the performance of the model on unseen data, and thus prevents overfitting from occurring.

To objectively evaluate the performance of the model, we used 5-fold cross validation. This method divides the dataset into multiple subsets, ensuring that the model can receive sufficient training and testing under different data combinations, thereby obtaining more robust and reliable performance indicators.

From Figure 4, it can be clearly seen that when the regularization parameter C value is small (0.001), the validation accuracy is low (average about 0.80), indicating a significant underfitting phenomenon; When C gradually increased to around 10, the accuracy of the model validation set reached its optimal level (about 0.90) and remained stable in each fold cross validation.

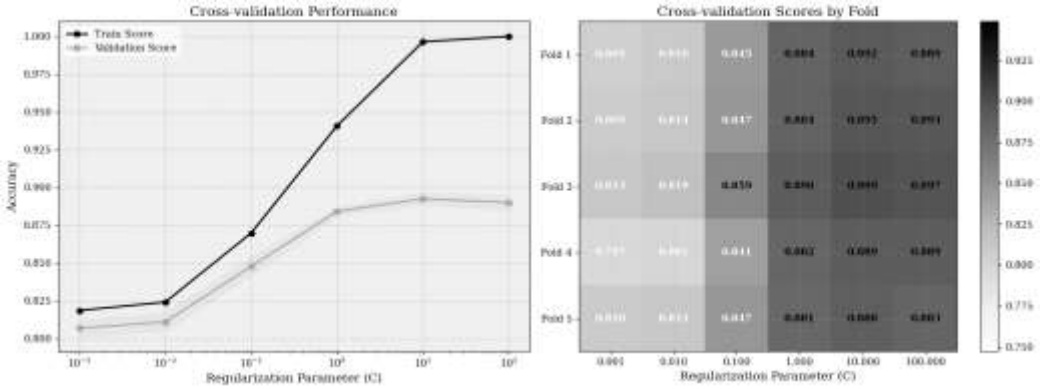


**Figure 4** Cross validation performance line chart and heat map

The visualized cross validation results not only help detect the fitting degree of the model on the training set, but also effectively reflect its predictive ability on unknown data.

### 2.5.3. Solver and Regularization

The choice of liblinear as the solver is based on its outstanding performance in large-scale text classification tasks. Liblinear can efficiently handle high-dimensional sparse features, and text classification tasks are typical high-dimensional sparse data scenarios. At the same time, it also performs well in terms of memory usage and computational efficiency, making it very suitable for big data processing in practical applications.

In terms of preventing model overfitting, we control the complexity of the model by adjusting the regularization parameter C. A smaller C value will enhance the regularization effect, thereby limiting the model's degrees of freedom and reducing the risk of overfitting; A larger C value allows the model to have better fitting performance on the training data.

From Figure 4, it can be further observed that when the C value exceeds 10, the accuracy of the training set approaches 100%, but the accuracy of the validation set no longer improves or even slightly decreases, indicating a slight overfitting tendency of the model in this region. Therefore, based on the results of cross validation, a C value of 10 was ultimately chosen, achieving a good balance between model fitting ability and generalization performance, and improving the overall robustness of the model.

### 2.5.4. Model compilation and training control

The entire training process is performed through a data classification model. The fit() function returns an object containing the training history. By analyzing the change curves of loss values and accuracy during the training process, we can intuitively determine whether the model has overfitting or underfitting phenomena and adjust hyperparameters in a timely manner.

This dynamic monitoring and tuning mechanism makes model training more flexible and efficient, while providing a solid theoretical basis and practical support for subsequent deployment.

## 2.6. Model evaluation and verification

### 2.6.1. Evaluation indicators

After completing the model training, we conducted a comprehensive performance evaluation of the model. The evaluation used multiple standard indicators, including Precision, Recall, and F1 score, and conducted in-depth analysis by combining ROC curve and Precision Recall curve (P-R curve).

The evaluation results on 25000 test samples show that the model exhibits good classification performance, as shown in Table 1.

**Table 1** Classification Performance

| category | Accuracy | recall | F1 |
|---|---|---|---|
| 0 | 0.87 | 0.89 | 0.88 |
| 1 | 0.89 | 0.87 | 0.88 |
| Avg | 0.88 | 0.88 | 0.88 |

Specifically, the model achieved a fairly balanced performance in the classification of negative comments (category 0) and positive comments (category 1).

For negative comments, the model achieved an accuracy of 0.87 and a recall of 0.89, with an F1 score of 0.88; For positive reviews, a precision of 0.89 and a recall of 0.87 were achieved, resulting in an F1 score of 0.88.

The overall accuracy of the model reaches 88%, which is a fairly good performance in text sentiment analysis tasks.

### 2.6.2. Confusion matrix

This article visualizes classification performance through confusion matrix to comprehensively understand the classification performance of the model on different categories, as shown in Figure 5, which is a confusion matrix diagram.

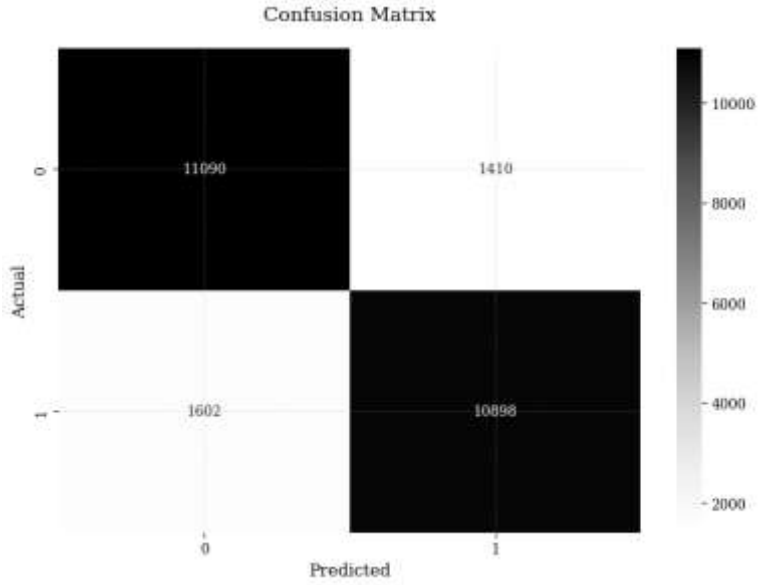


**Figure 5** Visual Confusion Matrix

The diagonal part of the matrix represents the number of correctly predicted samples, while the non diagonal part reflects the number of misclassified samples[7]. By analyzing the confusion matrix, we found that the model's predictions on positive and negative categories were relatively balanced, with a low misjudgment rate, further confirming the

effectiveness and robustness of the logistic regression model in text sentiment classification tasks.

### 2.6.3. Performance curve analysis

In order to gain a deeper understanding of the performance characteristics of the model, ROC curves and P-R curves were plotted and analyzed. The ROC curve is obtained by plotting the true rate versus false positive rate at different thresholds, and its mathematical expression is:

$$TPR = \frac{TP}{TP+FN} \tag{11}$$

$$FRP = \frac{FP}{FP+TN} \tag{12}$$

Among them, TP represents the number of correctly identified positive samples, FN represents the number of incorrectly identified negative samples, FP represents the number of incorrectly identified positive samples, and TN represents the number of correctly identified negative samples.

The area under the ROC curve (AUC) is an important indicator for evaluating a model's ability to distinguish between positive and negative samples, with an ideal AUC value of 1 and a randomly guessed AUC value of 0.5. Our model has a high AUC value, indicating its good classification and discrimination ability. The ROC curve is shown in Figure 6. It illustrates the performance of the logistic regression model across varying discrimination thresholds. The curve exhibits a rapid ascent towards the upper-left quadrant of the plot, indicating that a high True Positive Rate (TPR) is achieved at low False Positive Rates (FPR). The Area Under the AUC is documented as 0.95, a value that reflects the model's strong discriminatory power between positive and negative instances.

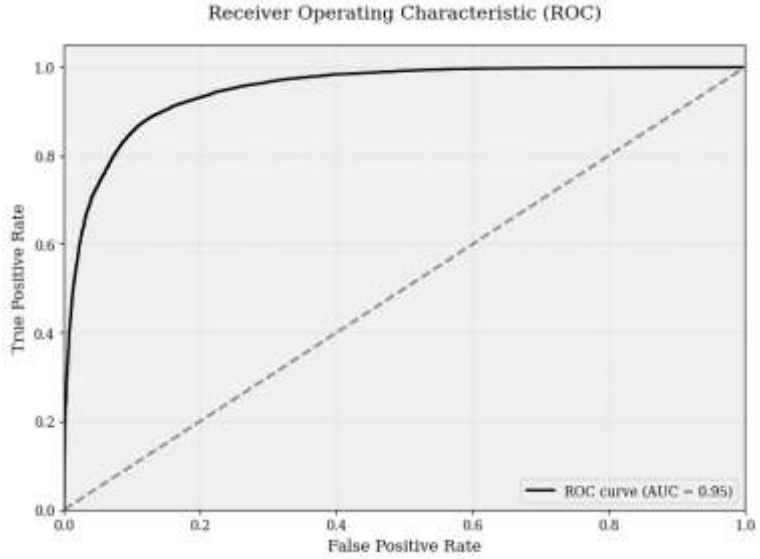


**Figure 6** ROC curve

Meanwhile, pay attention to the P-R curve, as shown in Figure 7. The Precision-Recall (P-R) curve delineates the model's precision in relation to varying levels of recall. It is observed that as recall increases, precision is maintained at a consistently high level, without a notable or abrupt decline. The AUC for this P-R curve is also recorded as 0.95, signifying that the model achieves a proficient balance between precision and recall.

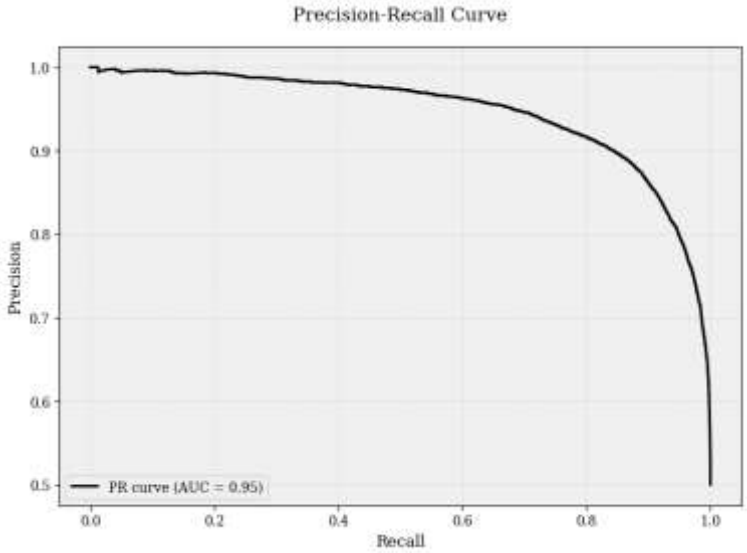


**Figure 7** Precision Recall Curve

The calculation formula for accuracy and recall is:

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

$$Recall = \frac{TP}{TP+FN} \quad (14)$$

Among them, TP represents the number of samples correctly predicted as positive by the model, FP represents the number of samples incorrectly predicted as positive by the model, and FN represents the number of samples incorrectly predicted as negative by the model[8].

The P-R curve demonstrates the trade-off between accuracy and recall of the model at different decision thresholds. Although the category distribution of our dataset is relatively balanced, valuable model performance information can still be obtained by observing the P-R curve, especially when adjusting the model prediction threshold is needed.

Through the comprehensive analysis of these evaluation indicators, it can be concluded that our logistic regression model has demonstrated stable and reliable performance in the task of movie review sentiment classification. The model not only performs well in overall accuracy, but also maintains high accuracy and recall when processing samples of different categories, indicating that the model has good generalization ability and practical value. Meanwhile, the higher ROC-AUC values and P-R curve performance further confirm the reliability of the model.

# 3. Conclusions

This research successfully culminated in the development of an integrated film review analysis system, demonstrating strong capabilities in both sentiment classification and thematic modeling. The study achieved notable quantitative outcomes: the Logistic Regression model, utilizing TF-IDF features, attained an accuracy of 88.0% on the test set, while the bidirectional LSTM model with Word2Vec embeddings reached 88.24%. More significantly, a strategic model fusion—combining Logistic Regression and LSTM predictions with a weighted average of 0.45:0.55 respectively, derived from validation set optimization—further elevated the sentiment classification accuracy to a superior 90.2%. Complementing this, Latent Dirichlet Allocation was effectively employed to categorize review texts into 15 distinct thematic groups, such as post-viewing sentiment, comedy, and social issues, enabling a multi-dimensional understanding of review content beyond simple polarity.

The core innovation of this work lies in its hybrid methodological approach, particularly the efficacy of the model fusion strategy in synergizing diverse algorithmic strengths for enhanced sentiment analysis. This research distinctively combines supervised learning for precise emotional polarity detection with unsupervised learning, through the application of LDA, to expand the analytical breadth towards thematic understanding. This dual approach facilitates a more comprehensive interpretation of film reviews, capturing not only the 'what' of sentiment but also the 'why' through contextual thematic insights, thereby creating a richer, more nuanced analytical output than achievable with singular methodologies.

Looking ahead, several strategic directions are envisioned for future development. Firstly, the adoption of more advanced neural architectures, such as Transformer-based models, could significantly enhance the capture of intricate linguistic nuances and long-range dependencies within review texts, potentially elevating classification and modeling accuracy further. Secondly, there is substantial potential in leveraging the established theoretical model—encompassing the fusion strategy and LDA-derived insights—as a foundational component for building sophisticated intelligent agents. By integrating these with large language models (LLMs), these agents could offer more dynamic, interactive, and context-aware analytical capabilities. Finally, the principles and system architecture developed herein possess considerable applicability in engineering and product development, paving the way for creating robust systems for broader text analytics, such as intelligent product review analysis platforms or automated customer feedback interpretation tools within various commercial ecosystems.

# References


[1] Zhang Yi. Research on Difference Detection Algorithm of UAV Aerial Images Based on Deep Learning[D]. Taiyuan: North University of China, 2021.

[2] Ji Siyang, Hou Kaihu. Research on Sentiment Analysis Considering Sentence Type Classification[J]. Software Guide, 2021, 20(10): 84–88.

[3] Xie Jinbao, Hou Yongjin, Kang Shouqiang, et al. Chinese Text Classification with Multi-Feature Fusion Based on Semantic Attention Neural Network[J]. Journal of Electronics and Information Technology, 2023, 40(5): 1258–1265.

[4] He Jianjun, Wang Xin, Gu Hong, et al. Multi-Instance Learning Algorithm Based on Logistic Regression Model and Cohesion Function[J]. Journal of Dalian University of Technology, 2020, (5): 788–793.

[5] Lü Shouqi, Hai Tao, Zheng Maoxing. Detection of Power Grid Network Attacks Combining CNN and LSTM[J]. Electronic Devices, 2023, 46(3): 824–830.

[6] Sun Jian, Liu Tonggan, Yang Hongwu, et al. Fault Diagnosis of Rolling Bearings under Variable Conditions Based on Improved CNN[J]. Thermal Power Engineering, 2025, 40(2): 176–186.

[7] Sun Jun, Zhu Weidong, Luo Yuanqiu, et al. Field Crop Leaf Disease Identification Based on Improved MobileNet-V2[J]. Transactions of the Chinese Society of Agricultural Engineering, 2021, 37(22).

[8] Han Wei, Xie Qing. Application of Key Artificial Intelligence Technologies in Power Customer Service[J]. Instrumentation Technology, 2024, (4): 1–3 + 12.