Journal of Deep Learning, Computer Vision and Digital Image Processing
E-ISSN: 2986-8939 P-ISSN: 2986-8920
Vol. 4, No. 2, June 2026
DOI. https://doi.org/10.61255/decoding.v4i2.1385

A Web-Based Sentiment Analysis System for IMDb Movie Reviews Using TF-IDF and Multinomial Naïve Bayes

Syahrur Ramadhan✉
(Universitas 'Aisyiyah Yogyakarta, Yogyakarta, Indonesia)

Sadr Lufti Mufreni
(Universitas 'Aisyiyah Yogyakarta, Yogyakarta, Indonesia)

OPEN ACCESS

ARTICLE HISTORY
Received: April 28, 2026
Revised: June 1, 2026
Accepted: June 27, 2026

KEYWORDS
Multinomial naïve bayes;
Natural language processing;
Sentiment analysis;
TF-IDF;
Web application deployment.

ABSTRACT

Purpose – The rapid growth of online movie platforms has produced large volumes of user reviews that contain valuable audience opinions. However, manual review analysis is inefficient. This study aims to develop a web-based sentiment analysis application using TF-IDF feature representation and the Multinomial Naïve Bayes algorithm to classify movie reviews into positive and negative sentiments.

Methods – The model was trained and evaluated using the IMDb 50K Movie Reviews dataset with an 80:20 train–test split. An additional 600 reviews from six different movies were used to demonstrate application-level implementation. Text preprocessing included cleaning, lowercase normalization, tokenization, stopword filtering, and lemmatization using Natural Language Processing techniques. The processed texts were transformed into TF-IDF vectors and classified using Multinomial Naïve Bayes with the default smoothing parameter ($\alpha = 1.0$). The trained model was deployed in a Flask-based web application for interactive sentiment prediction.

Findings – The model achieved an accuracy of 84.93%, with precision, recall, and F1-score showing relatively balanced performance across positive and negative classes. The web application successfully classified movie reviews and displayed sentiment distributions through an interactive interface.

Research implications – The findings indicate that lightweight machine learning methods can support practical web-based sentiment analysis with low computational demands. However, performance may decline when processing sarcasm, irony, or implicit contextual meaning.

Originality – This study combines benchmark evaluation with web-based validation using 600 additional real-world movie reviews, demonstrating practical applicability beyond dataset-level testing.

Correspondence Author: ✉ramadhansahrul239@gmail.com
To cite this article : Ramadhan, S., & Mufreni, S. L. (2026). A web-based sentiment analysis system for IMDb movie reviews using TF-IDF and Multinomial Naïve Bayes. Journal of Deep Learning, Computer Vision and Digital Image Processing, 4(2), 158-175. https://doi.org/10.61255/decoding.v4i2.1385

INTRODUCTION

The growing popularity of digital movie platforms has resulted in a large volume of user-contributed reviews being published on websites such as IMDb and Rotten Tomatoes. These reviews contain valuable information reflecting audience opinions, experiences, and satisfaction with movies, making them an important source of feedback for both viewers and content producers [1], [2]. However, the continuously growing volume of textual reviews makes manual analysis inefficient and time-consuming, particularly when timely insights into public opinion are required [3]. Consequently, Natural Language Processing (NLP) techniques have become widely adopted to automatically analyze textual information and identify sentiment patterns from large collections of user reviews [4], [5].

Sentiment analysis has attracted considerable attention in recent years because it enables researchers to derive meaningful insights from text data that are naturally unstructured. Various machine learning algorithms have been applied to sentiment classification tasks, among which Multinomial Naïve Bayes remains one of the most widely used approaches due to its computational efficiency and satisfactory performance on high-dimensional text data [6], [7]. In parallel, deep learning models such as Long Short-Term Memory (LSTM) have demonstrated promising capabilities in capturing contextual relationships among words and sentences, although these methods generally require greater computational resources and more complex training procedures [8], [9], [10]. Furthermore, advances in web technologies have also enabled sentiment analysis models to be deployed as interactive applications capable of producing interactive sentiment prediction [11].

Previous studies have reported competitive performance for Support Vector Machine (SVM) in classifying IMDb movie reviews [12]. Other research has shown that combining machine learning techniques with emotion analysis can further improve sentiment identification [13]. Comparative evaluations between Naïve Bayes and SVM indicate that the effectiveness of each algorithm depends on the characteristics of the dataset and the intended application [14]. Nevertheless, Multinomial Naïve Bayes continues to be an attractive choice because of its simplicity, scalability, and effectiveness when processing sparse textual representations [15]. These characteristics make it particularly suitable for sentiment classification tasks involving large volumes of online reviews [16].

Feature representation also plays an essential role in determining classification performance. Among various text representation methods, Term Frequency–Inverse Document Frequency (TF-IDF) is commonly employed to assign weights to terms according to their importance within documents and across the corpus [17]. Before feature extraction, preprocessing operations such as data cleaning, case folding, tokenization, stopword removal, and lemmatization are typically performed to reduce noise and improve text quality [18], [4]. Previous research has demonstrated that appropriate feature weighting and preprocessing strategies can substantially enhance classification accuracy [19]. Furthermore, the combination of TF-IDF and Multinomial Naïve Bayes has consistently produced reliable results in sentiment analysis applications involving opinion mining and textual reviews [20], [21].

Although previous studies have demonstrated that both machine learning and deep learning techniques can achieve promising performance in movie review sentiment classification [12], [13], [14], [15], most of them primarily emphasize predictive accuracy or algorithm comparison. Several studies have also implemented sentiment analysis within Flask-based web applications; however, these implementations commonly rely on computationally intensive deep learning models or focus on specific analytical objectives, such as emotion detection or genre-based analysis, with limited attention to lightweight deployment, implementation reproducibility, and practical software evaluation [11], [16]. Consequently, there remains a need for a web-based sentiment analysis framework that combines efficient text representation, computationally efficient classification, systematic software implementation, and practical deployment while maintaining reliable predictive performance.

Based on this research gap, this study develops a web-based sentiment analysis system for English movie reviews by integrating TF-IDF feature representation with the Multinomial Naïve Bayes algorithm within a Flask framework. Unlike many previous studies that primarily emphasize model evaluation, this research combines sentiment classification, practical software implementation, functional testing, and application deployment within a single operational framework. The model was trained and evaluated using the IMDb 50K Movie Reviews dataset and further demonstrated using 600 manually collected IMDb reviews. The proposed framework is expected to provide an efficient and reproducible reference for lightweight web-based sentiment analysis applications.

The remainder of this paper is organized as follows. Section II presents the methodology. Section III discusses the experimental results and system implementation. Finally, Section IV concludes the study and outlines future work.

METHOD

Research Design

This research adopted the System Development Life Cycle (SDLC) framework as the overall software development methodology for developing a web-based sentiment analysis system for IMDb movie reviews. This study employs the SDLC framework because the research aims not only to construct a sentiment classification model but also to transform the trained model into an operational web application. Since the objective of this study includes both machine learning model development and software implementation, SDLC was adopted to manage the complete software engineering process rather than only the predictive modeling process. In this study, the SDLC framework served as a structured guideline for organizing the research workflow, encompassing literature review, data collection, data processing, model development, model evaluation, and system implementation. Therefore, SDLC was employed to ensure that each development stage was conducted systematically from problem identification to application deployment [22].

Within the proposed SDLC workflow, machine learning techniques were incorporated during the data processing, model development, and evaluation activities before the trained model was integrated into the web application. The proposed system employs the Multinomial Naïve Bayes algorithm to classify movie reviews into positive and negative sentiment categories. Prior to classification, the review texts underwent several Natural Language Processing (NLP) procedures, including text cleaning, lowercase normalization, tokenization, stopword removal, and lemmatization. The processed texts were subsequently transformed into Term Frequency–Inverse Document Frequency (TF-IDF) feature vectors before model training [3], [4], [5]. After the classification model was trained and evaluated, it was deployed using the Flask framework to provide interactive sentiment prediction through a web-based interface [6].

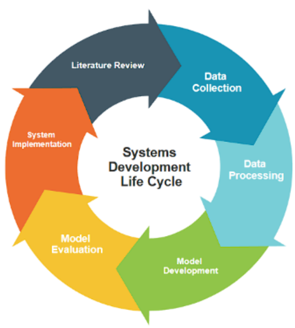

Figure 1. Research workflow implemented within the SDLC framework.

Figure 1. illustrates the research workflow implemented within the adopted SDLC framework. The process begins with a literature review to identify research gaps and define system requirements, followed by data collection and preprocessing, model development using TF-IDF and Multinomial Naïve Bayes, model evaluation, and finally implementation of the trained model into a Flask-based web application [22].

Table 1. Research Workflow within the SDLC Framework

Workflow Stage	Activities Performed in This Study
Literature Review	Review previous studies, identify research gaps, define research objectives, and determine system requirements.
Data Collection	Collect the IMDb 50K Movie Reviews dataset for model development and an additional 600 manually collected IMDb reviews for application-level demonstration.
Data Processing (Preprocessing + TF-IDF)	Perform text cleaning, case folding, tokenization, stopword removal, lemmatization, and TF-IDF feature extraction.
Multinomial Naïve Bayes (Model Development)	Train the classifier using the training dataset and serialize the trained model and TF-IDF vectorizer for deployment.
Model Evaluation	Evaluate the trained model using the testing dataset through confusion matrix analysis, accuracy, precision, recall, F1-score, five-fold stratified cross-validation, and computational performance measurement.
System Implementation	Integrate the trained model into a Flask-based web application, implement the user interface, perform functional testing, and deploy the application in a local environment.

Table 1. summarizes the research activities performed throughout the SDLC-based development process. The SDLC served as the overall software engineering framework for organizing the development of the proposed web-based sentiment analysis application. Within this framework, the machine learning pipeline including text preprocessing, TF-IDF feature extraction, model development, model evaluation, and deployment was integrated into the software development lifecycle to ensure that both the classification model and the application were systematically developed and evaluated [22].

Literature Review

As the initial activity within the proposed SDLC-based research workflow, a literature review was conducted to identify previous studies related to sentiment analysis, TF-IDF, Multinomial Naïve Bayes, and web-based sentiment analysis systems [5]. This stage also identified limitations in earlier research, particularly studies that primarily focused on model development without integrating API-based services into practical applications [7]. The findings obtained from this review served as the foundation for defining the system requirements, formulating the research problem, and establishing the objectives of the proposed web-based sentiment analysis system.

Data Collection

This study employed two complementary datasets to support both sentiment classification model development and application-level evaluation. The primary dataset was the IMDb 50K Movie Reviews dataset, which contains 50,000 English-language movie reviews labeled as either positive or negative. This publicly available benchmark dataset was used for training and evaluating the sentiment classification model [1], [13]. In addition, a supplementary dataset consisting of 600 movie reviews was collected to assess the practical performance of the developed web-based application under real-world conditions [6].

The IMDb 50K Movie Reviews dataset was selected because it is one of the most widely used benchmark datasets for sentiment analysis research. Its balanced distribution of positive and negative reviews enables objective evaluation of classification models while facilitating comparison with previous studies employing the same benchmark dataset [13], [14]. For model development, the dataset was divided into 80% training data and 20% testing data using the train_test_split() function with random_state = 42 to ensure reproducibility throughout the model training and evaluation process.

To complement the benchmark evaluation, an additional dataset comprising 600 user reviews was manually collected from publicly available user review pages on the IMDb website between October and November 2025. The reviews were obtained from six selected movies, namely The Dark Knight, Cats (2019), Barbie (2023), Joker (2019), The Emoji Movie, and Spirited Away, with 100 randomly selected reviews collected from each movie. These six movies were intentionally selected to represent diverse sentiment characteristics, including predominantly positive, predominantly negative, and relatively balanced audience responses. The reviews were manually collected without utilizing an Application Programming Interface (API) or automated web-scraping techniques. Unlike the IMDb 50K dataset, the additional reviews were not used for model training. Since these reviews did not contain ground-truth sentiment labels, they were not used to calculate quantitative evaluation metrics such as accuracy, precision, recall, or F1-score. Instead, they were employed to assess the practical implementation of the developed web-based application by demonstrating its ability to classify previously unseen movie reviews under real-world usage conditions [6].

All datasets were stored in CSV and XLSX file formats to facilitate data organization, preprocessing, feature extraction, model development, and integration with the Flask-based web application. The structured dataset organization also supported reproducibility by enabling consistent data processing throughout the entire research workflow.

All movie reviews used in this study were obtained from publicly accessible sources on the IMDb platform. The collected reviews were utilized exclusively for academic research purposes and were analyzed in aggregated form without attempting to identify, contact, or disclose any personal information related to individual users. Consequently, the research did not involve human participants or require access to private or confidential data.

Data Processing

Before feature extraction and sentiment classification, all movie reviews underwent a series of preprocessing procedures to improve data consistency and reduce textual noise. The preprocessing pipeline consisted of data cleaning, case folding, tokenization, stopword removal, and lemmatization. These procedures were implemented in Python 3.12 using the Natural Language Toolkit (NLTK version 3.9.1) for tokenization, stopword removal, and lemmatization, BeautifulSoup4 (version 4.13.5) for HTML tag removal, and Python's Regular Expression (re) library for text normalization [4], [5], [17].

Data Cleaning

HTML tags were removed using BeautifulSoup, while numbers, punctuation, emojis, and special characters were eliminated using regular expression (regex) operations. This process reduced textual noise and generated cleaner review texts for subsequent preprocessing stages [4], [17].

Table 2. Data Cleaning

Before	After
"Amazing movie!!! 😍 😍 10/10 #bestmovieever"	"Amazing movie best movie ever"

Case Folding

Case folding standardized text by converting all characters into lowercase, ensuring that words differing only in letter case were treated as identical features during classification [18].

Table 3. Case Folding

Before	After
"Amazing Movie"	"amazing movie"

Tokenization

Each review was segmented into individual words using the word_tokenize() function provided by NLTK, producing tokens used for feature extraction and sentiment classification [18].

Table 4. Tokenization

Before	After
"amazing Movie"	["amazing", "movie"]

Stopword Removal

Common English stopwords with limited semantic contribution were removed using the NLTK stopword corpus, reducing feature dimensionality while preserving informative terms for sentiment classification [4], [18].

Table 5. Stopword Removal

Before	After
"this","is","an","amazing","movie"	"amazing movie"

Lemmatization

The remaining tokens were normalized using WordNetLemmatizer, which converts inflected words into their base forms to reduce vocabulary variation and improve feature consistency [4], [18].

Table 6. Lemmatization

Before	After
"movies", "amazing", "watching"	"movie", "amazing", "watch"

From an implementation perspective, these preprocessing procedures standardized the textual data before TF-IDF feature extraction, producing more consistent feature representations for sentiment classification using the Multinomial Naïve Bayes algorithm [4], [17], [18]. No explicit negation handling strategy was applied during preprocessing. Negation terms such as "not" and "never" were preserved as independent tokens and represented through TF-IDF weighting.

The identical preprocessing pipeline was consistently applied during both the model training phase and the web-based prediction process. Applying the same sequence of preprocessing operations ensured that the textual features generated during deployment remained compatible with those learned by the trained classification model.

TF–IDF Feature Extraction

Following the preprocessing stage, the cleaned review texts were transformed into numerical feature vectors using the Term Frequency–Inverse Document Frequency (TF-IDF) weighting scheme. TF-IDF emphasizes informative terms by assigning higher weights to words that frequently occur within a document while reducing the influence of terms that appear across many documents, thereby generating more representative features for sentiment classification [5], [19].

In this study, the TfidfVectorizer implementation provided by the Scikit-learn library(version 1.6.1) was employed with max_features = 5000, while the remaining parameters were kept at their default values (min_df = 1, max_df = 1.0, and ngram_range = (1,1)). This configuration retained up to 5,000 informative textual features for model training. The vectorizer was fitted exclusively on the training dataset using the fit_transform() function, whereas the testing dataset was transformed using the transform() function. This approach prevented information leakage from the testing dataset and ensured an unbiased evaluation of the classification model [5].

The unigram configuration (ngram_range = (1,1)) was selected to maintain a compact feature space and reduce computational complexity while preserving the lightweight characteristics of the proposed web-based application. Although higher-order n-grams may capture additional contextual information, unigram features have been widely adopted in Multinomial Naïve Bayes text classification because they provide an effective balance between predictive performance and computational efficiency [5], [17].

Although unigram features cannot explicitly capture polarity inversion in expressions such as "not good", they were Intentionally adopted to preserve computational efficiency and maintain a lightweight deployment architecture.

The resulting sparse TF-IDF feature matrix served as the numerical representation of the review texts and was subsequently used as input for training and evaluating the Multinomial Naïve Bayes classifier.

$$\boldsymbol{TF(t,d)} = \frac{\boldsymbol{f_{t,d}}}{\boldsymbol{\sum_k f_{k,d}}} \quad (1)$$

$$\boldsymbol{DF(t)} = \frac{\boldsymbol{N_t}}{\boldsymbol{N}} \quad (2)$$

$$\boldsymbol{IDF(t)} = \mathbf{log}\left(\frac{\boldsymbol{N}}{\boldsymbol{DF(t)}}\right) \quad (3)$$

$$\boldsymbol{TF\text{-}IDF(t,d)} = \boldsymbol{TF(t,d)} \times \boldsymbol{IDF(t)} \quad (4)$$

Model Development (Multinomial Naïve Bayes)

The TF-IDF feature vectors generated during the feature extraction stage were subsequently used as input for the Multinomial Naïve Bayes classifier. The IMDb 50K Movie Reviews dataset was divided into 80% training data and 20% testing data using the train_test_split() function with a fixed random state of 42 to ensure reproducibility during model development and evaluation. The classifier was implemented using the MultinomialNB algorithm from the Scikit-learn library with the default smoothing parameter (α = 1.0). After the training process was completed, both the trained classification model and the TF-IDF vectorizer were serialized and stored as .pkl files using the Joblib library to facilitate integration with the Flask-based web application [3], [6], [8].

Multinomial Naïve Bayes applies Bayes' theorem to estimate the probability that a document belongs to a particular sentiment class based on the extracted textual features. The posterior probability is formulated as follows:

$$\boldsymbol{P(C|d)} = \frac{\boldsymbol{P(d|C)P(C)}}{\boldsymbol{P(d)}} \quad (5)$$

Since (P(d)) remains constant for all classes, the classification process can be simplified as:

$$\boldsymbol{P(C|d)} \propto \boldsymbol{P(d|C)}\ \boldsymbol{P(C)} \quad (6)$$

Assuming conditional independence among words (the naïve assumption), the probability of a document given a class is calculated as:

$$\boldsymbol{P(d|C)} = \prod_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{P(w_i|C)} \quad (7)$$

For the Multinomial Naïve Bayes model, the conditional probability of each word is computed using Laplace smoothing:

$$\boldsymbol{P(w_i|C)} = \frac{\boldsymbol{N_{w_{i,C}} + 1}}{\boldsymbol{N_C + |V|}} \quad (8)$$

Model Evaluation

The performance of the proposed sentiment classification model was evaluated using the testing dataset through a confusion matrix and several standard classification metrics, namely accuracy, precision, recall, and F1-score. These evaluation metrics were computed using the Scikit-learn evaluation library by comparing the predicted sentiment labels with the corresponding ground-truth labels [12], [13].

Table 7. Confusion Matrix

	Predicted Positive	**Predicted Negative**
Actual Positive	True Positive	False Negative
Actual Negative	False Positive	True Negative

The confusion matrix summarizes the numbers of TP, TN, FP, and FN, which form the basis for assessing the classification performance of the proposed model.

Accuracy:

$$\frac{TP+TN}{TP+TN+FP+FN} \quad (9)$$

Precision:

$$\frac{TP}{TP+FP} \quad (10)$$

Recall:

$$\frac{TP}{TP+FN} \quad (11)$$

F1-Score:

$$2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (12)$$

System Implementation

The final stage involved deploying the trained model through a Flask API by integrating the serialized Multinomial Naïve Bayes model (model_naive_bayes.pkl) and the TF-IDF vectorizer (vectorizer_tfidf.pkl) into the application backend using the Joblib library [21]. The web application accepts movie reviews as user input, performs the same preprocessing and TF-IDF transformation applied during model training, and subsequently generates sentiment predictions in JSON format [16]. To enhance usability, the user interface was developed using HTML, CSS, and JavaScript, enabling users to perform sentiment analysis through a web-based interface [6]. The application was deployed and evaluated in a local development environment using the Flask framework, where all prediction processes were executed locally without relying on external hosting services or cloud-based infrastructure. The deployment environment consisted of a laptop equipped with an AMD Ryzen 5 PRO 4650U processor, 16 GB RAM, running Python 3.12 with Flask 3.1.0, Scikit-learn 1.6.1, NLTK 3.9.1, and BeautifulSoup4 4.13.5.

RESULTS AND DISCUSSION

Results

A sentiment classification model was constructed using the Multinomial Naïve Bayes algorithm with TF-IDF feature representation. Model training and testing were conducted on the IMDb 50K Movie Reviews dataset using an 80:20 data split. Furthermore, 600 reviews obtained from six selected movies were utilized to assess the implementation of the web-based application.

Table 8. Model Evaluation Results

Metric	Value
Accuracy	84.93%
Precision	0.85
Recall	0.85
F1-Score	0.85

Table 8. presents the evaluation results of the proposed model, which achieved an accuracy of 84.93%, with precision, recall, and F1-score values of 0.85. The evaluation results indicate balanced precision, recall, and F1-score values across both sentiment classes.

Table 9. Classification Report

Class	Precision	Recall	F1-Score	Support
Negative	0.85	0.84	0.85	4961
Positive	0.85	0.85	0.85	5039

Table 9. summarizes the classification results for the positive and negative sentiment classes. The similarity of the precision, recall, and F1-score values across both classes suggests that the model performs consistently when handling different sentiment categories. The classification report demonstrates that the proposed model achieved relatively balanced performance across both sentiment categories. The negative sentiment class obtained a recall value of 0.84, while the positive class achieved 0.85. The similarity of these values indicates that the classifier does not exhibit substantial bias toward a particular sentiment category. The classification report shows similar precision, recall, and F1-score values for both sentiment classes.

A confusion matrix was used to further assess classification performance. The results show that most reviews were correctly classified, although a small number of prediction errors remained.

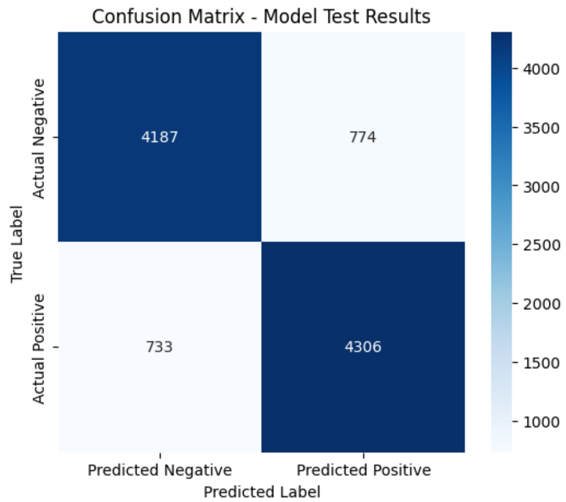

Figure 2. Confusion Matrix

Figure 2. shows that the majority of review instances were correctly classified into their respective sentiment categories. The confusion matrix shows that most review instances were correctly classified, while a smaller proportion were assigned to incorrect sentiment classes. Based on the confusion matrix, the proposed model correctly classified 4,306 positive reviews (True Positive) and 4,187 negative reviews (True Negative). Meanwhile, 774 negative reviews were incorrectly classified as positive (False Positive), and 733 positive reviews were incorrectly classified as negative (False Negative). These results indicate that the classifier successfully identified the majority of review sentiments while maintaining a relatively balanced number of misclassification errors across both sentiment classes.

To further assess the robustness of the proposed model, five-fold stratified cross-validation was performed using the IMDb 50K Movie Reviews dataset. As presented in Table 10, the proposed Multinomial Naïve Bayes classifier achieved a mean accuracy of 0.8522 with a standard deviation of 0.0035. The corresponding mean precision, recall, and F1-score were 0.8480, 0.8582, and 0.8531, respectively. The relatively small standard deviations across all evaluation metrics indicate that the proposed model maintained stable classification performance under different data partitions.

Table 10. Five-Fold Stratified Cross-Validation Results

Fold	Accuracy	Precision	Recall	F1-Score
1	0.8470	0.8441	0.8512	0.8476
2	0.8566	0.8520	0.8632	0.8575
3	0.8531	0.8471	0.8618	0.8546
4	0.8510	0.8464	0.8576	0.8520
5	0.8532	0.8503	0.8574	0.8538
Mean ± Std	0.8522 ± 0.0035	0.8480 ± 0.0031	0.8582 ± 0.0047	0.8531 ± 0.0036

The cross-validation results demonstrate that the proposed model consistently achieved similar classification performance across different folds. Although the primary evaluation based on an 80:20 train-test split yielded an accuracy of 84.93%, the five-fold cross-validation produced a comparable mean accuracy of 85.22%. The relatively small standard deviations across all evaluation metrics further indicate that the proposed model maintained consistent predictive performance across different data partitions.

The relatively small standard deviations across all evaluation metrics indicate that the proposed classifier maintained stable performance under different data partitions. This suggests that the reported classification results were consistent and were not strongly influenced by a particular train-test split.

After model deployment, the classification model was deployed within a Flask-based web application. System demonstration using 600 movie reviews demonstrated varying sentiment distributions across different films. Movies such as *The Dark Knight* and *Spirited Away* were dominated by positive sentiment, whereas *Cats* and *The Emoji Movie* received predominantly negative sentiment. In contrast, *Barbie* and *Joker* exhibited relatively balanced sentiment distributions.

System Implementation Analysis

Following model training, the classifier was integrated into a Flask-based web application to demonstrate practical sentiment analysis using 600 manually collected IMDb reviews. The application successfully processed user inputs, generated sentiment predictions, and visualized sentiment distributions across the selected movies, demonstrating the operational feasibility of the proposed framework in a web environment

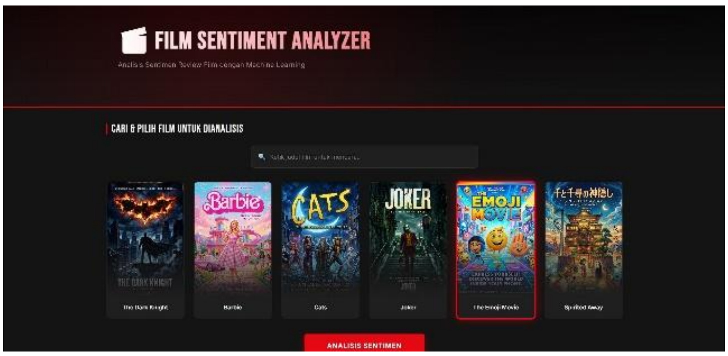

Figure 3. Main Interface

Figure 3 illustrates the main interface of the developed application. The homepage provides access to movie selection and sentiment analysis features through a simple web interface designed to improve usability.

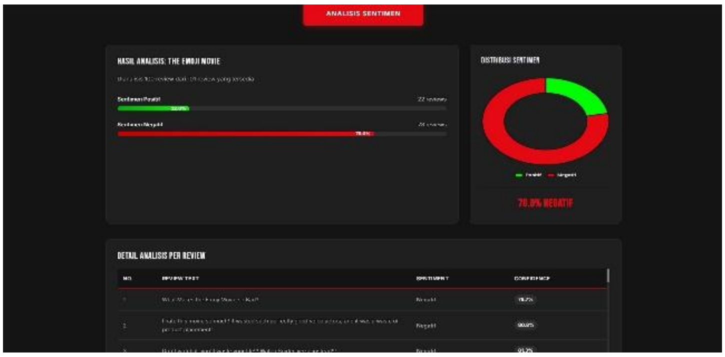

Figure 4. Sentiment Analysis Results

Figure 4 presents the sentiment analysis results generated by the application after preprocessing, TF-IDF transformation, and classification. The interface summarizes the predicted positive and negative sentiment distributions for the selected movie reviews.

Figure 5. Sentiment Distribution Across Movies

Figure 5. compares sentiment distributions across the six evaluated movies. The visualization provides a clear overview of audience responses by presenting the proportions of positive and negative reviews for each film. Such visual representation allows users to identify sentiment trends more efficiently than manual review analysis and may assist in understanding public opinion regarding specific movie titles.

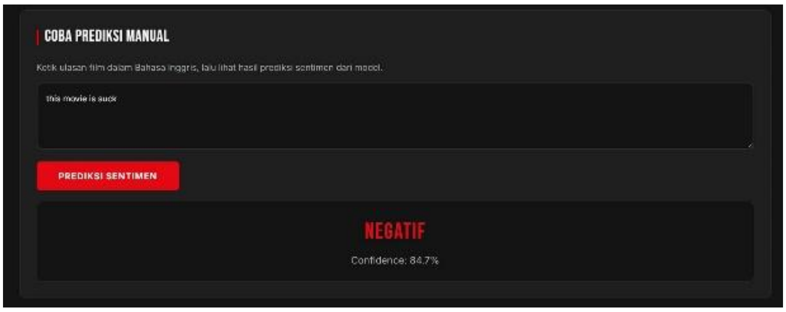

Figure 6. Sentiment Prediction Interface

Figure 6. illustrates the review prediction interface used for interactive sentiment prediction. The interface allows users to enter movie reviews and obtain sentiment predictions through an interactive web-based environment. This functionality demonstrates the practical applicability of the proposed framework and highlights how sentiment analysis can be delivered as an operational service rather than remaining solely as an experimental model.

The developed application integrates text preprocessing, TF-IDF feature extraction, and Multinomial Naïve Bayes classification within a single Flask-based environment. Submitted reviews are processed automatically through the prediction pipeline, and sentiment results are presented interactively through the web interface. These findings demonstrate that lightweight machine learning models can be effectively deployed for practical sentiment analysis while maintaining responsive prediction performance.

Overall, the implementation results demonstrate that the proposed system can effectively represent public opinion through a web-based sentiment analysis platform. Furthermore, integrating the machine learning model into a Flask API provides opportunities for future applications in other domains, including product reviews and social media analysis [20], [21]. The successful deployment of the model also confirms that lightweight machine learning approaches can be integrated into practical applications while maintaining satisfactory predictive performance and efficient response process.

To complement the classification performance evaluation, computational performance was also assessed by measuring the prediction latency of the deployed web application. The latency measurement was conducted in the local deployment environment by processing 100 movie reviews through the Flask API while recording the elapsed processing time required for sentiment prediction. This evaluation was performed to examine the responsiveness of the developed application during practical use.

Table 11. Deployment Environment and Latency Measurement

Metric	Value
Deployment Environment	Local Flask API
Processor	AMD Ryzen 5 PRO 4650U
Memory	16 GB RAM
Dataset Processed	100 movie reviews
Average Prediction Time	29.84 ms

Table 11. summarizes the deployment environment and latency measurement of the developed web-based sentiment analysis application. The experiment was conducted by processing 100 movie reviews using the locally deployed Flask API. The developed application processed 100 movie reviews with an average prediction time of 29.84 ms in the evaluated local deployment environment. These results indicate that the proposed TF-IDF and Multinomial Naïve Bayes framework provides responsive prediction performance while maintaining low computational overhead, supporting its suitability for interactive web-based sentiment analysis.

Functional Testing

To verify that the developed web-based sentiment analysis application operated correctly, black-box functional testing was performed on all major application features. The evaluation covered movie search, movie selection, sentiment analysis, visualization, manual prediction, comparison functionality, and input demonstration. The testing focused on verifying whether each system function produced the expected output from the user's perspective.

Table 12. Functional Testing Results

No	Function	Test Scenario	Expected Result	Actual Result	Status
1	Homepage	Open the web application	Homepage and movie selection interface are displayed successfully	Homepage displayed correctly	Pass
2	Movie Search	Enter a movie title in the search box	Matching movie is displayed according to the entered keyword	Search results displayed correctly	Pass
3	Movie Selection	Select one of the six available movies	Selected movie is highlighted and ready for analysis	Movie selected successfully	Pass

No	Function	Test Scenario	Expected Result	Actual Result	Status
4	Movie Sentiment Analysis	Click the Analyze Sentiment button	System analyzes the selected movie reviews and displays sentiment statistics	Sentiment analysis completed successfully	Pass
5	Sentiment Distribution Visualization	Analyze a selected movie	Positive and negative sentiment distribution chart is displayed correctly	Chart displayed successfully	Pass
6	Review-Level Sentiment Classification	Analyze a selected movie	Individual reviews with predicted sentiment labels and confidence scores are displayed	Review predictions displayed correctly	Pass
7	Movie Comparison	Select two or more movies and click Compare	Comparative sentiment chart is generated for the selected movies	Comparison chart displayed successfully	Pass
8	Manual Sentiment Prediction	Enter a custom movie review and click Predict Sentiment	System predicts the sentiment of the entered review	Prediction displayed successfully	Pass
9	Empty Input Handling	Submit an empty manual review	Warning message is displayed and prediction is not executed	Warning message displayed correctly	Pass
10	Model Loading	Start the Flask application	The serialized Multinomial Naïve Bayes model and TF-IDF vectorizer are loaded successfully	Model and vectorizer loaded successfully	Pass

Table 12. summarizes the functional testing results of the developed web application. All evaluated features operated according to the expected system requirements. All evaluated functionalities operated according to the expected system behavior, indicating that the integration between the trained classification model and the Flask-based application was successfully implemented.

The testing results demonstrate that the integration between the Flask-based web application and the trained Multinomial Naïve Bayes model was successfully implemented. User inputs were processed correctly through the preprocessing pipeline, TF-IDF feature extraction, and sentiment prediction stages before being presented through an interactive web interface. These findings indicate that the developed application is functionally reliable for demonstrating web-based sentiment analysis of movie reviews.

Discussion

The proposed framework achieved an accuracy of 84.93%, demonstrating that TF-IDF feature representation combined with the Multinomial Naïve Bayes algorithm can effectively classify English movie reviews while maintaining low computational complexity suitable for lightweight web-based deployment.

The balanced precision, recall, and F1-score values observed across both sentiment classes further indicate that the classifier did not exhibit a strong tendency to favor either positive or negative predictions. This balanced behavior is important for practical sentiment analysis because both classes can be identified with comparable reliability.

The effectiveness of the proposed model is largely attributed to TF-IDF weighting, which emphasizes informative terms while reducing the influence of less relevant words. This representation improves feature quality and enables balanced sentiment classification without requiring computationally intensive deep learning models [5].

The obtained accuracy is consistent with previous studies employing Multinomial Naïve Bayes and TF-IDF for sentiment classification [14], [15]. Although transformer-based and Support Vector Machine approaches generally achieve higher predictive performance [10], [12], [17], they typically require greater computational resources. In contrast, the proposed framework offers a practical trade-off between classification accuracy, computational efficiency, and deployment simplicity for lightweight web-based sentiment analysis.

Table 13. Direct Comparison with Previous MNB-Based Studies

Study	Dataset	Method	Accuracy	Precision	Recall	F1-Score	Main Contribution
[14]	IMDb 50k	Multinomial NB + TF-IDF	0.8764	0.8879	0.8600	0.8737	Optimized TF-IDF-based Multinomial Naïve Bayes for IMDb sentiment classification
[15]	IMDb 50k	Multinomial NB	0.8617	0.8613	0.8599	0.8606	Comparative evaluation of multiple machine learning classifiers
This Study	IMDb 50k	TF-IDF + Multinomial NB + Flask	0.8493	0.8500	0.8500	0.8500	Lightweight web deployment and application-level demonstration using 600 manually collected IMDb reviews

Table 13 shows that although the proposed framework achieved slightly lower accuracy than previous Multinomial Naïve Bayes studies, it maintained balanced precision, recall, and F1-score values while extending previous work through deployment as a Flask-based web application. The additional application-level demonstration using 600 manually collected IMDb reviews further highlights that practical web deployment can be achieved without substantially compromising classification performance.

Table 14. Contextual Comparison with Alternative Approaches

Study	Method	Accuracy	Precision	Recall	F1-Score	Main Contribution
[12]	SVM + TF-IDF + NRCLex Emotion Detection	0.9000	0.9000	0.9000	0.9000	Integration of emotion detection with sentiment classification
[15]	Linear SVM	0.8948	0.8882	0.9016	0.8949	Performance comparison of traditional ML classifiers
[16]	RoBERTa + CNN + Flask	0.9150	0.9200	0.9200	0.9100	Deep learning with contextual embeddings and Interactive prediction Flask deployment

Table 14. shows that although several machine learning and deep learning approaches achieved higher classification performance, the proposed framework offers a practical balance between predictive performance, computational efficiency, and deployment simplicity. Rather than focusing

solely on benchmark evaluation, this study complements previous work by implementing the trained classifier within a Flask-based web application for interactive sentiment prediction.

To better understand the classification errors, a qualitative analysis was conducted on 75 randomly selected misclassified reviews from the IMDb 50K testing dataset. Most errors occurred in reviews containing mixed sentiment, where positive and negative opinions appeared simultaneously, making the dominant sentiment difficult to determine. Additional errors were associated with long review structures, context-dependent expressions, and ambiguous wording, which reduced the effectiveness of TF-IDF in representing the overall semantic meaning. Only a few errors involved explicit negation. These findings suggest that the primary limitation of the proposed TF-IDF and Multinomial Naïve Bayes framework lies in its bag-of-words representation, which does not explicitly capture contextual relationships between words.

This observation reflects a common limitation of frequency-based text classification methods. Although TF-IDF effectively identifies important terms, it does not preserve word order or contextual relationships between words [18]. Consequently, reviews containing nuanced or context-dependent expressions remain challenging to classify accurately, encouraging the adoption of contextual language models in recent sentiment analysis research.

Another limitation of the proposed framework is the handling of negation expressions. Although words such as not and never were retained during preprocessing, they were treated as independent features rather than as contextual modifiers. As a result, reviews containing complex negation patterns or implicit contextual meanings may still lead to incorrect sentiment predictions. Representative examples of the identified misclassification patterns are presented in Table 15 to illustrate the characteristics of reviews that challenged the proposed classifier.

Table 15. Representative Examples of Misclassified Reviews

Review Snippet	Actual	Predicted	Possible Cause
"...this is not a masterpiece, but a good old-fashioned scary movie... that's rather rare nowadays."	Positive	Negative	Mixed Sentiment
"...the movie would not be completely great... however, I agree one hundred percent..."	Positive	Negative	Context-dependent expression
"...I'm glad they got rid of the rest of the team... the writing, acting and animation are top notch..."	Positive	Negative	Long and complex review

The successful deployment of the classifier within a Flask-based application demonstrates its practical applicability beyond benchmark evaluation. The system enables users to analyze movie reviews and visualize sentiment distributions through an interactive web interface [16], [21]. From a scientific perspective, the results confirm that TF-IDF and Multinomial Naïve Bayes remain effective for movie review sentiment classification when supported by appropriate preprocessing and feature engineering [10], [17]. From a practical perspective, the developed Flask-based application enables users to perform interactive sentiment analysis through an accessible web interface, demonstrating the feasibility of deploying lightweight machine learning models for real-world opinion mining. Overall, the proposed framework provides a practical balance between classification performance, computational efficiency, and deployment simplicity.

Implications

The findings demonstrate that lightweight machine learning techniques can provide reliable web-based sentiment analysis while requiring relatively modest computational resources. The proposed framework therefore offers a practical solution for resource-constrained environments where efficient sentiment prediction is required.

Beyond movie reviews, the developed application may be adapted for product reviews, customer feedback analysis, and social media opinion mining. From an academic perspective, the integration of NLP preprocessing, TF-IDF feature extraction, Multinomial Naïve Bayes classification, and Flask deployment provides a complete reference framework for future sentiment analysis research.

Research Contribution

This study provides two main contributions. First, it demonstrates that the integration of TF-IDF feature representation and the Multinomial Naïve Bayes algorithm can achieve reliable sentiment classification performance for English movie reviews while maintaining relatively low computational complexity. The findings further support the continued applicability of conventional machine learning methods for text classification tasks, particularly in resource-constrained environments.

Second, this study extends previous work by implementing the trained sentiment classification model as a functional Flask-based web application. Unlike studies that primarily focus on model evaluation, the proposed framework integrates text preprocessing, feature extraction, sentiment prediction, and result visualization into a single operational system capable of providing interactive sentiment prediction analysis. This implementation demonstrates the practical applicability of machine learning models beyond experimental evaluation and provides a reference framework for future web-based sentiment analysis applications.

Limitations

Despite the encouraging results, several limitations remain. The current study is restricted to English-language movie reviews and only considers binary sentiment categories. In addition, the classification approach relies on statistical word representations, which may limit its ability to accurately interpret contextual expressions such as sarcasm, irony, and implicit meanings. The evaluation was also conducted using a single benchmark dataset and a limited collection of movie reviews. Therefore, the generalizability of the proposed model to other domains, languages, and sentiment classification settings requires further investigation.

Suggestions

Future work may focus on incorporating more advanced language modeling techniques, including transformer-based architectures and deep learning approaches, to improve contextual understanding. Expanding the dataset to include multilingual reviews and additional sentiment categories may further enhance model generalization. Moreover, the integration of explainable artificial intelligence methods could improve the transparency and interpretability of prediction results.

CONCLUSION

A web-based sentiment analysis system for English movie reviews was designed and implemented in this study by combining TF-IDF weighting and Multinomial Naïve Bayes classification. Evaluation using the IMDb 50K Movie Reviews dataset produced an accuracy of 84.93%, with balanced precision, recall, and F1-score values. The results suggest that the developed model can effectively distinguish positive and negative sentiments while remaining computationally efficient.

The implementation of the trained model through a Flask API further demonstrated its practical applicability by enabling interactive sentiment prediction in a web environment. Demonstration using 600 reviews from six different movies confirmed that the developed application could effectively present sentiment distributions and support interactive analysis of user opinions.

Despite these promising results, the current approach remains limited in handling reviews containing complex contextual expressions such as sarcasm and irony. Future research may investigate transformer-based or deep learning models, incorporate multilingual datasets, and explore explainable artificial intelligence techniques to further improve contextual understanding and prediction interpretability.

ACKNOWLEDGMENT

The authors sincerely appreciate all individuals and institutions that supported the completion of this research. Special thanks are extended to the providers of the IMDb 50K Movie Reviews dataset, which served as the primary source of data for model training and evaluation. The authors also acknowledge Universitas 'Aisyiyah Yogyakarta for its academic support throughout this study.

AUTHOR CONTRIBUTION STATEMENT

SR conceived the research idea, collected and processed the data, developed the sentiment analysis model, implemented the web-based application, conducted the experiments, analyzed the results, and prepared the manuscript. SM supervised the research, reviewed the methodology, demonstrated the findings, provided academic guidance, and revised the manuscript.

AI DISCLOSURE STATEMENT

OpenAI ChatGPT was utilized during manuscript preparation to assist with language refinement, translation, and editorial enhancement. All generated content was carefully reviewed, verified, and revised by the author where necessary. The author assumes full responsibility for the accuracy, originality, and integrity of the final manuscript.

REFERENCES

[1] A. N. Zhafira, N. Afifah, S. A. Putri, V. Marhalatun, D. Intan, and S. Saputra, "Analisis sentimen ulasan film pada IMDb menggunakan algoritma Naive Bayes," *JUMISTIK*, vol. 4, no. 1, pp. 373–383, 2025, doi: 10.70247/jumistik.v4i1.139.

[2] M. S. Islam *et al.*, "Challenges and future in deep learning for sentiment analysis: a comprehensive review and a proposed novel hybrid approach," *Artif. Intell. Rev.*, vol. 57, no. 3, pp. 1–79, 2024, doi: 10.1007/s10462-023-10651-9.

[3] N. S. Fathullah, Y. A. Sari, and P. P. Adikara, "Analisis sentimen terhadap rating dan ulasan film dengan menggunakan metode klasifikasi Naïve Bayes dengan fitur lexicon-based," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 4, no. 2, pp. 590–593, 2020.

[4] N. Malik and M. Bilal, "Natural language processing for analyzing online customer reviews: a survey, taxonomy, and open research challenges," *PeerJ Comput. Sci.*, vol. 10, p. e2203, 2024, doi: 10.7717/peerj-cs.2203.

[5] K. P. Harmandini and K. Muslim, "Analysis of TF-IDF and TF-RF feature extraction on product review sentiment," *Sink. J. dan Penelit. Tek. Inform.*, vol. 8, no. 2, pp. 929–937, 2024, doi: 10.33395/v8i2.13376.

[6] Ismaturrahmi, M. Khadafi, and Amirullah, "Implementation of web-based sentiment analysis application on movie reviews using the Naïve Bayes algorithm," *J. Informatics Eng. Softw. Appl.*, vol. 1, no. 1, pp. 129–138, 2025.

[7] Martiti and C. Juliane, "Implementation of Naïve Bayes algorithm on sentiment analysis application," *Adv. Eng. Res.*, vol. 207, pp. 193–200, 2021.

[8] J. Lu, H. Fan, and Y. Zhang, "Sentiment analysis of IMDb movie reviews based on LSTM," *J. Adv. Eng. Technol.*, vol. 2, no. 2, pp. 1–11, 2025, doi: 10.62177/jaet.v2i2.429.

[9] A. Gupta, S. Pandey, M. P. Behera, S. Darshana, and A. Dash, "Sentiment analysis of movie review using machine learning," *Educ. Adm. Theory Pract.*, vol. 29, no. 4, pp. 1724–1734, 2024, doi: 10.53555/kuey.v29i4.6616.

[10] S. V. Kanse, O. A. Desai, and U. R. Pol, "Sentiment analysis of IMDb movie review using deep learning," *Int. Res. J. Mod. Eng. Technol. Sci.*, vol. 7, no. 4, pp. 1033–1039, 2025.

[11] Suwarno, Wesly, and B. Syahputra, "Genre-based sentiment and emotion system for audience insight," *TIERS Inf. Technol. J.*, vol. 6, no. 2, pp. 217–240, 2025, doi: 10.38043/tiers.v6i2.7183.

[12] A. Hermawan, R. Yusuf, B. Daniawan, and Junaedi, "Enhanced sentiment analysis and emotion detection in movie reviews using support vector machine algorithm," *TELKOMNIKA Telecommun. Comput. Electron. Control*, vol. 23, no. 1, pp. 138–146, 2025, doi: 10.12928/TELKOMNIKA.v23i1.26377.

[13] D. Subedi, N. Lamichhane, and N. Subedi, "Sentiment analysis of IMDb movie reviews using SVM and Naïve Bayes classifier," *J. Eng. Sci.*, vol. 4, no. 1, pp. 1–13, 2025, doi: 10.3126/jes2.v4i1.70138.

[14] C. Dewi, R. Chen, H. J. Christanto, and F. Cauteruccio, "Multinomial Naïve Bayes classifier for sentiment analysis of Internet Movie Database," *Vietnam J. Comput. Sci.*, vol. 10, no. 4, pp. 485–498, 2023, doi: 10.1142/S2196888823500100.

[15] M. M. Danyal, S. S. Khan, M. Khan, M. B. Ghaffar, B. Khan, and M. Arshad, "Sentiment analysis based on performance of Linear Support Vector Machine and Multinomial Naïve Bayes using movie reviews with baseline techniques," *J. Big Data*, vol. 5, pp. 1–18, 2023, doi: 10.32604/jbd.2023.041319.

[16] B. Paneru, B. Thapa, and B. Paneru, "Sentiment analysis of movie reviews: a Flask application using CNN with RoBERTa embeddings," *Syst. Soft Comput.*, vol. 7, p. 200192, 2025, doi: 10.1016/j.sasc.2025.200192.

[17] Y. Mamani-Coaquira and E. Villanueva, "A review on text sentiment analysis with machine learning and deep learning techniques," *IEEE Access*, vol. 12, pp. 193115–193130, 2024, doi: 10.1109/ACCESS.2024.3513321.

[18] P. Monika, C. Kulkarni, H. N. Kumar, S. Shruthi, and V. Vani, "Machine learning approaches for sentiment analysis: a survey," *Int. J. Health Sci. (Qassim).*, vol. 6, no. S4, pp. 1286–1300, 2022, doi: 10.53730/ijhs.v6nS4.6119.

[19] H. Zhao, Z. Liu, X. Yao, and Q. Yang, "A machine learning-based sentiment analysis of online product reviews with a novel term weighting and feature selection approach," *Inf. Process. Manag.*, vol. 58, no. 5, p. 102656, 2021, doi: 10.1016/j.ipm.2021.102656.

[20] F. J. Damanik and D. B. Setyohadi, "Analysis of public sentiment about COVID-19 in Indonesia on Twitter using Multinomial Naïve Bayes and Support Vector Machine," *IOP Conf. Ser. Earth Environ. Sci.*, vol. 704, no. 1, p. 012027, 2021, doi: 10.1088/1755-1315/704/1/012027.

[21] T. M. A. Admira, Sutarno, and M. Saefudin, "Model sentiment analysis berbasis machine learning untuk data Genz-Career Aspiration menggunakan Flask dan Naïve Bayes," *J. Inf. Syst. Informatics Comput.*, vol. 9, no. 1, pp. 25–39, 2025, doi: 10.52362/jisicom.v9i1.1880.

[22] R. S. Pressman and B. R. Maxim, *Software Engineering: A Practitioner's Approach*, 9th ed. New York, NY: McGraw-Hill Education, 2019.