Movie Review Classifier

Md Asif Tanvir, Md Abdur Rahman Fahad, and Abiha Tahsin Chowdhury

Department of Computer Science, Missouri State University, Springfield, USA

{mt5864s@missouristate.edu, mf8494s@MissouriState.edu,
ac444s@MissouriState.edu}

I. Project Description

With the continuous growth of online movie reviews, classifying the reviews automatically to a specific category is becoming an essential need for the users. In this project, we have implemented a movie rating classifier that will predict whether a movie review is positive or negative using information retrieval techniques. Using existing review data that are labeled as "positive" or "negative," we want to build an automated system that will be able to classify a new movie review as a "positive" or "negative" review.

For this project, we have conducted an experiment with a document classification method, Naive Bayes, to classify different positive and negative reviews. We have measured and analyzed the classification model's performance using common metrics like accuracy, precision, recall, and F1 score to make sure the final system is both dependable and effective. Finally, we have run experiments with different feature selection methods like Mutual Information, χ^2 feature selection, and Frequency-based feature selection (Collection Frequency) to determine which method performs the best with Naive Bayes for our case.

II. Literature Review

As people write more online reviews, especially about movies, it becomes important to understand what they are saying—whether they liked the movie or not. This task is called sentiment classification. It sounds simple, but it's not. Unlike topic classification, where the subject is clear, sentiment is often hidden in tone, sarcasm, or context. For example, “How could anyone sit through this movie?” has no clearly negative words, but it’s still a bad review. Polarity classification [1] helps audiences quickly gauge public sentiment [2]. Numerous studies have proposed models for this task.

A pioneering study showed that sentiment classification differs significantly from topic classification [3]. The researchers used machine learning methods like Naive Bayes [4], SVM [5], and Maximum Entropy [6] to classify reviews. Their best result came from using SVM with unigrams, which reached 82.9% accuracy. This work became a starting point for many later studies.

Building on this, another study explored ensemble classifiers combining SVM, Maximum Entropy, and score-based methods. Their system worked better and reached 87.5% accuracy. This showed that using multiple models together can improve results [7]. In 2016, a five-class system was used instead of just positive or negative. They used n-grams, SentiWordNet, and several classifiers. Among them, Random Forest worked best, giving almost 89% accuracy. But their method depended a lot on a predefined lexicon, which may not work well in other domains [8].

A more focused comparative study implemented NB and SVM using Amazon book reviews [9]. Using TF-IDF [10] for feature extraction, they found that SVM slightly outperformed NB (84% vs. 82.87%). That reaffirms the importance of feature representation in traditional models.

Some studies have explored unsupervised methods for classifying movie reviews. One study compared a supervised method that used word patterns (n-grams) with an unsupervised method based on word meanings. The supervised method performed better, reaching around 85% accuracy, while the unsupervised one reached 77%. However, the unsupervised method had the advantage of being faster and not needing labeled data, making it helpful in real-time situations [11].

In another study, researchers used both Naive Bayes and Markov Models [12] to classify reviews. They experimented with filtering the text using part-of-speech tags and word groupings from a dictionary. Surprisingly, the filters didn’t improve results on larger datasets and sometimes even made them worse [13]. Language differences were also studied. In one case, researchers worked on Chinese movie reviews and used a method that was adjusted for the language. The accuracy was lower—around 74%—but it showed that models often need to be adapted to the language they're working with [14].

Some recent approaches returned to Naive Bayes and compared it with CNN [15]. One study classified reviews into three categories—positive, negative, and neutral—using a Bag-of-Words model and achieved 72.8% accuracy, which was better than CNN in terms of speed [16]. However, it still struggled with subtle expressions and lacked any feature selection technique. Meanwhile, another method used a hybrid feature selection technique combining TF-IDF and SVM-RFE [17]. This helped select the most important features before applying SVM for classification [18]. A different study tested several classifiers—SVM, decision trees, gradient boosting, and random forests on IMDB reviews using a range of feature extraction techniques like TF-IDF, BoW, GloVe [19], and Word2Vec [20]. The best performance came from combining SVM with TF-IDF and BoW using the TextBlob tool [21].

More recently, deep learning methods have been tested. One approach used a deep neural network with several layers on IMDB reviews [22]. It worked well on the dataset it was trained on, but didn't do as well on other datasets. This showed that deep learning models can sometimes overfit and not generalize to new data. In another domain, healthcare, a method was developed that used sentence structure and deep learning to classify doctor reviews [23]. Even though it wasn't focused on movie reviews, it showed how combining grammar-based analysis with neural networks can be powerful.

Even with all these improvements, many challenges remain. Sarcasm, mixed feelings, and negation are still hard to detect. Some methods rely too much on predefined word lists, which may not work across different topics. Others work well only on specific datasets. Also, there hasn't been enough focus on using feature selection methods like Mutual Information [24] or χ^2 [25], which could improve simpler models. We focus on a Naive Bayes—based implementation and we plan to test different ways of picking the most useful features. Our goal is to build a system that is simple, effective, and easy to apply to new data.

III. Background

For implementing this project, we had to rely on multiple tools. First of all, for normalizing the texts, we have used the *NLTK* library from Python [26]. This is used for tokenizing the training data texts, removing the stop words, and stemming the texts for later use.

IV. Methodology

For implementing this review classification, we have to go through a step-by-step process. The details for each step can be found below:

A. Data Preprocessing

For data preprocessing, we have first loaded the data from the text file in lowercase and applied tokenization using the *NLTK* library. Then, the stop words have been removed from the list of the tokens. We have used the stop words list from the *NLTK* library. Then finally, the tokens are stemmed using *Porter Stemmer*.

B. Naive Bayes Implementation

As the text classification method, we have chosen Naive Bayes. Naive Bayes classifies documents into predetermined types based on the likelihood of a word occurring by using the concepts of the Bayes theorem. The formula is given as equation 1:

$$P(C_k|\mathbf{x}) = \log P(C_k) + \sum_{i=1}^{n} \log P(x_i|C_k) \tag{1}$$

Where: C_k is the class label (e.g., *positive* or *negative*) and $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ are the tokenized words in a review. We implemented $P(C_k)$ and $P(x_i|C_k)$ from the list of tokenized words we gathered from the preprocessing step.

C. Feature Selections

For better performance and optimization, feature selection is usually done in Naive Bayes. Feature selection is a process of selecting a subset of features from the total pool of features which are more important in the classification. In this case, the features would be the words from the reviews. In feature selection method, we will mainly select the top performing words for each class and use them in Naive Bayes implementation. We have implemented and used 3 different feature selection methods. The details for these methods are given below:

- **Mutual Information**: Mutual Information calculates how much information is contributed by a words presence or absence. The higher the score is, the bigger the contribution of that term. For implementing this, we have used equation 2:

$$I(U;C) = \frac{N_{11}}{N}\,\log_2\!\Big(\frac{N\,N_{11}}{N_{1\cdot}\,N_{\cdot 1}}\Big) + \frac{N_{01}}{N}\,\log_2\!\Big(\frac{N\,N_{01}}{N_{0\cdot}\,N_{\cdot 1}}\Big) + \frac{N_{10}}{N}\,\log_2\!\Big(\frac{N\,N_{10}}{N_{1\cdot}\,N_{\cdot 0}}\Big) + \frac{N_{00}}{N}\,\log_2\!\Big(\frac{N\,N_{00}}{N_{0\cdot}\,N_{\cdot 0}}\Big) \tag{2}$$

  Where:

$$N = N_{11} + N_{10} + N_{01} + N_{00},$$

$$N_{1\cdot} = N_{11} + N_{10}, \quad N_{0\cdot} = N_{01} + N_{00},$$

$$N_{\cdot 1} = N_{11} + N_{01}, \quad N_{\cdot 0} = N_{10} + N_{00},$$

$$N_{11} = \big|\{\, d : U \in d \;\wedge\; C(d) = \text{positive}\}\big|,$$

$$N_{10} = \big|\{\, d : U \in d \;\wedge\; C(d) = \text{negative}\}\big|,$$

$$N_{01} = \big|\{\, d : U \notin d \;\wedge\; C(d) = \text{positive}\}\big|,$$

$$N_{00} = \big|\{\, d : U \notin d \;\wedge\; C(d) = \text{negative}\}\big|,$$

- χ^2 **(Chi-Square)**: χ^2 provides the information on the independence of two events: occurrence of a term, and occurrence of a class. For implementing this, we have used equation 3.

$$\chi^2 = \frac{N\left(N_{11}\,N_{00} - N_{10}\,N_{01}\right)^2}{(N_{11} + N_{10})\,(N_{11} + N_{01})\,(N_{10} + N_{00})\,(N_{01} + N_{00})} \tag{3}$$

Where:

$$N = N_{11} + N_{10} + N_{01} + N_{00},$$

$$N_{11} = \big|\{\, d : U \in d \;\wedge\; C(d) = \text{positive}\}\big|,$$

$$N_{10} = \big|\{\, d : U \in d \;\wedge\; C(d) = \text{negative}\}\big|,$$

$$N_{01} = \big|\{\, d : U \notin d \;\wedge\; C(d) = \text{positive}\}\big|,$$

$$N_{00} = \big|\{\, d : U \notin d \;\wedge\; C(d) = \text{negative}\}\big|,$$

- **Collection Frequency**: Another feature selection method is selecting the features based on frequency. Here, different frequencies could be used based on the use case, like term frequency, document frequency, or collection frequency. For our project, we have chosen collection frequency as the feature selection method. Collection frequency refers to the total number of times a word occurs in a class. Equation 4 has been used for calculating this.

$$\mathrm{CF}(u) = \sum_{d=1}^{N} \mathrm{tf}_d(u) \tag{4}$$

Where:

$$\mathrm{tf}_d(u) = \text{the raw count of term } u \text{ in document } d.$$

D. Data

For this project, we have used a labeled dataset that contains the reviews in text format along with the respective sentiment for that review. The dataset has a total of 50000 movie reviews labeled as “Positive” or “Negative.”

V. Results

A. Evaluation Metrics

For evaluating the model, we have divided our dataset into a 90:10 ratio of training and testing data. The 90% training data is then trained using the Naive Bayes model we developed. Then, we have used different evaluation metrics, like precision, recall, and F1 score, to evaluate the model using the test data. The output is shown in Table I.

TABLE I
PERFORMANCE METRICS OF NAIVE BAYES CLASSIFIER

Class	**Precision**	**Recall**	**F1-Score**
Negative	0.84	0.88	0.86
Positive	0.88	0.84	0.86
Accuracy		**0.8584**	

B. Analysis on Different Feature Selection Methods

To analyze the effects of the 3 different feature selection methods, we ran our Naive Bayes model using these selected features. First, we determined the top features for each of these methods. Then, we used these top features from each method with Naive Bayes and determined the accuracy and the F1 scores. To better understand the effects of these feature selection methods, we iterated through multiple numbers of features selected. In Table II, we have shown these results for varying number of features selected.

TABLE II
ACCURACY AND F1 SCORES FOR EACH FEATURE-SELECTION METHOD AT VARYING NUMBERS OF SELECTED FEATURES

No. of Features	**Mutual Information**		**χ^2 (Chi-Square)**		**Frequency**	
	Acc	**F1**	**Acc**	**F1**	**Acc**	**F1**
10	0.7350	0.7687	0.7350	0.7687	0.5576	0.5567
100	0.8276	0.8277	0.8294	0.8288	0.7280	0.7293
500	0.8468	0.8459	0.8476	0.8466	0.8218	0.8219
1 000	0.8510	0.8497	0.8512	0.8501	0.8280	0.8272
5 000	0.8486	0.8442	0.8464	0.8422	0.8420	0.8391
10 000	0.8468	0.8421	0.8454	0.8407	0.8440	0.8394
20 000	0.8482	0.8428	0.8482	0.8428	0.8488	0.8439

In Figure 1, we demonstrate how the F1 score behaves for the number of features selected by each method. From this, we can see that the score increases as the number of features increases to a point for both mutual information and χ^2. The scores reach their peak when around 1000 terms are selected as features and decrease from there. For collection frequency, however, the score is consistently lower than that of the other two methods, and it kept increasing with the number of features increasing. From these results, we can conclude that both mutual information and χ^2 preformed better than the frequency-based method, and the optimal number of features to be selected for classification is 1000.

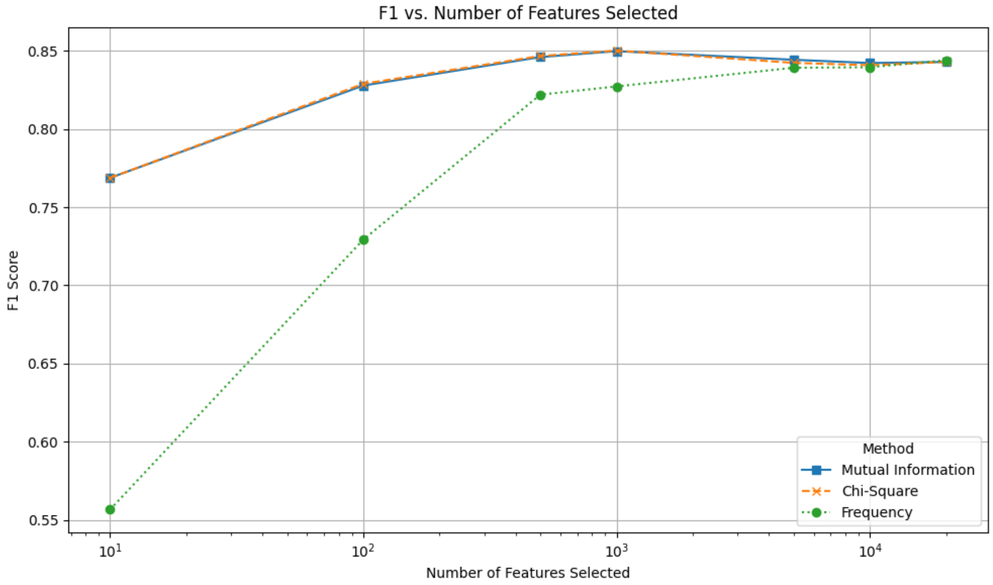

Fig. 1. Accuracy and weighted-F1 scores vs. number of selected features for the three feature-selection methods.

VI. Conclusion

This project set out to build a simple yet effective movie review classifier using a Naive Bayes model combined with different feature selection techniques. By experimenting with Mutual Information, χ^2, and Collection Frequency, we were able to demonstrate how the right features can significantly boost performance. Both Mutual Information and χ^2 outperformed the frequency-based approach, especially when selecting around 1,000 key features—proving that smarter feature selection really does matter. Our classifier reached an accuracy of over 85%, with balanced precision and recall, making it a reliable solution for basic sentiment classification tasks. This lightweight model is easy to train, fast to run, and surprisingly competitive. Moving forward, there's a lot of exciting potential in combining traditional models with modern techniques like word embeddings or hybrid approaches to handle trickier cases like sarcasm or mixed opinions.

References

[1] "Polarity classification," https://www.sciencedirect.com/topics/computer-science/polarity-classification, accessed: 2025-04-16.

[2] M. Raees and S. Fazilat, “Lexicon-based sentiment analysis on text polarities with evaluation of classification models,” *arXiv preprint arXiv:2409.12840*, 2024. [Online]. Available: https://arxiv.org/abs/2409.12840

[3] B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up? sentiment classification using machine learning techniques,” in *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2002. [Online]. Available: https://arxiv.org/abs/cs/0205070

[4] G. I. Webb, E. Keogh, and R. Miikkulainen, “Naïve bayes.” *Encyclopedia of machine learning*, vol. 15, no. 1, pp. 713–714, 2010.

[5] S. Suthaharan and S. Suthaharan, “Support vector machine,” *Machine learning models and algorithms for big data classification: thinking with examples for effective learning*, pp. 207–235, 2016.

[6] N. Wu, *The maximum entropy method*. Springer Science & Business Media, 2012, vol. 32.

[7] K. Tsutsumi, K. Shimada, and T. Endo, “Movie review classification based on a multiple classifier,” in *The 21st Pacific Asia Conference on Language, Information and Computation: Proceedings*, vol. 21. Waseda University, 2007, pp. 481–488.

[8] T. P. Sahu and S. Ahuja, “Sentiment analysis of movie reviews: A study on feature selection & classification algorithms,” in *2016 International Conference on Microelectronics, Computing and Communications (MicroCom)*. Ieee, 2016, pp. 1–6.

[9] S. Dey, S. Wasif, D. S. Tonmoy, S. Sultana, J. Sarkar, and M. Dey, “A comparative study of support vector machine and naive bayes classifier for sentiment analysis on amazon product reviews,” in *2020 International Conference on Contemporary Computing and Applications (IC3A)*. IEEE, 2020, pp. 217–220.

[10] A. Aizawa, “An information-theoretic perspective of tf–idf measures,” *Information Processing & Management*, vol. 39, no. 1, pp. 45–65, 2003.

[11] P. Chaovalit and L. Zhou, “Movie review mining: A comparison between supervised and unsupervised classification approaches,” in *Proceedings of the 38th annual Hawaii international conference on system sciences*. IEEE, 2005, pp. 112c–112c.

[12] L. Rabiner and B. Juang, “An introduction to hidden markov models,” *ieee assp magazine*, vol. 3, no. 1, pp. 4–16, 1986.

[13] F. Salvetti, S. Lewis, and C. Reichenbach, “Automatic opinion polarity classification of movie reviews,” *Colorado research in linguistics*, 2004.

[14] Q. Ye, W. Shi, and Y. Li, “Sentiment classification for movie reviews in chinese by improved semantic oriented approach,” in *Proceedings of the 39th annual Hawaii international conference on system sciences (HICSS’06)*, vol. 3. IEEE, 2006, pp. 53b–53b.

[15] K. O’shea and R. Nash, “An introduction to convolutional neural networks,” *arXiv preprint arXiv:1511.08458*, 2015.

[16] S. Gowri, R. Surendran, M. Divya Bharathi, and J. Jabez, “Improved sentimental analysis to the movie reviews using naive bayes classifier,” in *2022 International Conference on Electronics and Renewable Systems (ICEARS)*. IEEE, 2022, pp. 1831–1836.

[17] H. Sanz, C. Valim, E. Vegas, J. M. Oller, and F. Reverter, “Svm-rfe: selection and visualization of the most relevant features through non-linear kernels,” *BMC bioinformatics*, vol. 19, pp. 1–18, 2018.

[18] N. S. M. Nafis and S. Awang, “An enhanced hybrid feature selection technique using term frequency-inverse document frequency and support vector machine-recursive feature elimination for sentiment classification,” *Ieee Access*, vol. 9, pp. 52 177–52 192, 2021.

[19] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation,” in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[20] K. W. Church, “Word2vec,” *Natural Language Engineering*, vol. 23, no. 1, pp. 155–162, 2017.

[21] M. Z. Naeem, F. Rustam, A. Mehmood, I. Ashraf, G. S. Choi *et al.*, “Classification of movie reviews using term frequency-inverse document frequency and optimized machine learning algorithms,” *PeerJ Computer Science*, vol. 8, p. e914, 2022.

[22] K. Ullah, A. Rashad, M. Khan, Y. Ghadi, H. Aljuaid, and Z. Nawaz, “A deep neural network-based approach for sentiment analysis of movie reviews,” *Complexity*, vol. 2022, no. 1, p. 5217491, 2022.

[23] R. Rivas, N. Montazeri, N. X. Le, and V. Hristidis, “Automatic classification of online doctor reviews: evaluation of text classifier algorithms,” *Journal of medical Internet research*, vol. 20, no. 11, p. e11141, 2018.

[24] P. E. Latham and Y. Roudi, “Mutual information,” *Scholarpedia*, vol. 4, no. 1, p. 1658, 2009.

[25] T. M. Franke, T. Ho, and C. A. Christie, “The chi-square test: Often used and more often misinterpreted,” *American journal of evaluation*, vol. 33, no. 3, pp. 448–458, 2012.

[26] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit*. O’Reilly Media, Inc., 2009.