Comparison Of Neural Network and Traditional Classifiers for Twitter Sentiment Analysis

Haolan Guo*

Faculty of Engineering, The University of Sydney, Sydney, NSW2006, Australia

* Corresponding Author Email: hguo4658@uni.sydney.edu.au

Abstract. Sentiment analysis has been a popular topic of study in the field of social media analysis, particularly when it comes to analyzing the emotions expressed in online comments. This is particularly relevant when it comes to IMDb Movie reviews, where users often express their opinions on the films they have watched. By using sentiment analysis techniques, researchers can gain insights into the overall sentiment of a movie and how it is perceived by the public. This information can be useful for movie studios and producers, as it can help them gauge the success of their films and make decisions about future productions. In this analysis, the performance of neural network models is compared with that of traditional classification methods when applied to the task of sentiment classification of tweets. A dataset of tweets collected from IMDb movie reviews is used for training. Three different models are trained on this dataset: a sequential neural network with two dense layers activated by ReLU and SoftMax functions, logistic regression, and random forest. The performance of these models is evaluated using a variety of metrics, including confusion matrices, AUC graphs, and accuracy and loss curves. It is found that the neural network model achieves an accuracy of approximately 90%, outperforming the logistic regression and random forest models, which achieve accuracies of approximately 90% and 83%, respectively.

Keywords: Deep learning; machine learning; semantic analysis; neural network.

1. Introduction

Sentiment analysis is a research field within the broader domain of natural language processing [1, 2]. It involves the use of computational methods to analyze the emotions and attitudes expressed in written or spoken language. It is relevant in the context of social media, where users often express their opinions on various topics, including films, TV shows, and other forms of entertainment [3, 4]. Sentiment analysis of IMDb Movie reviews is a valuable tool for researchers, movie studios, and other stakeholders in the entertainment industry. It can provide valuable insights into the public's perception of a film and can be used to inform decision-making and future productions [5].

One of the key applications of sentiment analysis is in the analysis of IMDb Movie reviews [6, 7]. IMDb is one of the most popular websites for movie fans, with millions of users around the world providing ratings and reviews for films. By using sentiment analysis techniques, researchers can gain insights into the overall sentiment of a movie and how it is perceived by the public. For example, a study published in the Journal of Information Science found that sentiment analysis of IMDb Movie reviews could be used to predict the box office success of films [8]. The study analyzed the sentiment of over 100,000 IMDb reviews for 50 popular films and found that movies with higher average ratings and more positive reviews tended to have higher box office revenues.

In addition to predicting box office success, sentiment analysis of IMDb Movie reviews can also be used to identify key themes and trends within a particular film. For example, a study published in the International Journal of Human-Computer Studies used sentiment analysis to identify common themes in the reviews of the popular TV show "Game of Thrones" [9]. The study found that the most common themes were related to the characters, plot, and overall enjoyment of the show.

Sentiment analysis involves categorizing comments as either positive or negative in order to understand and manage sentiment related to a specific topic. This technique is useful for evaluating people's feelings about a brand, product, or domain. It is used in various industries, including product marketing, healthcare, finance and hospitality, in order to gather opinions and allows people to understand the perspective of targeted groups based on text data.

2. Method

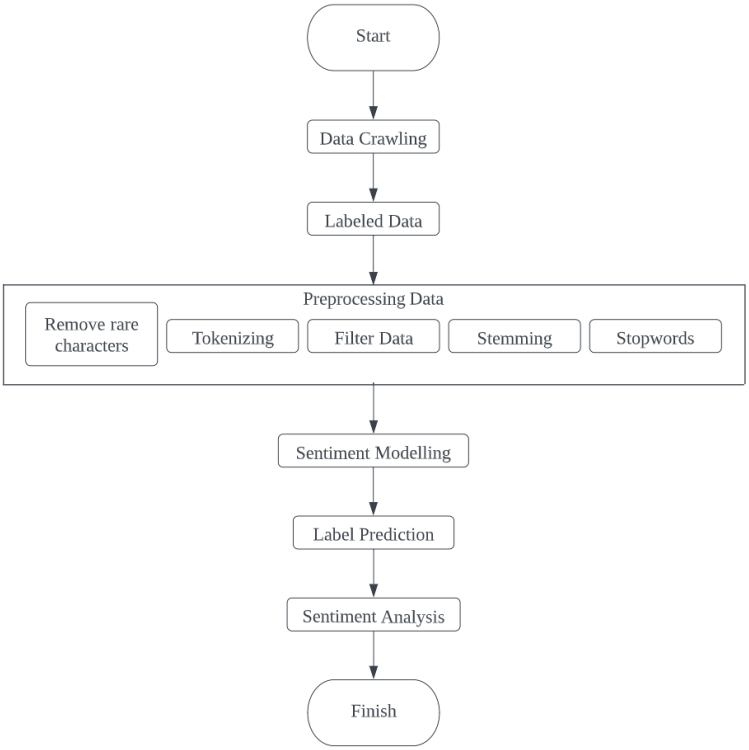

Fig. 1 Workflow of this method.

The research process is depicted in Figure 1, which illustrates the workflow for this analysis.

2.1. Dataset

The data [10] used in this study was from IMDb. There are 40,000 movie reviews labeled as either positive (1) or negative (0). It contains data for training, validation, and testing. The training set allows the model to learn patterns and relationships in the data, the validation set helps to optimize the model's hyperparameters, and the test set enables us to assess the model's generalization ability.

	text	label
0	I grew up (b. 1965) watching and loving the Th...	0
1	When I put this movie in my DVD player, and sa...	0
2	Why do people who do not know what a particula...	0
3	Even though I have great interest in Biblical ...	0
4	Im a die hard Dads Army fan and nothing will e...	1

Fig. 2 Examples of the dataset.

The features on the movie comment data can be seen in Figure 2. The number of each type of comment is almost balanced. More specifically, there are 20019 negative comments and 19981 positive comments, thus no balancing algorithms is needed to apply to this data.

2.2. Data pre-processing

To label the video comments data with 0 or 1 to indicate positive or negative sentiment, TextBlob, a Python text processing toolkit with a natural language processing (NLP) API, is used. Before the data can be further processed, it needs to be cleaned to remove unrelated letters, numbers, and symbols. The data is then subjected to several preprocessing stages, including:

1) Removing rare characters to improve the time efficiency and accuracy of classification.

2) Tokenization, which involves converting text data into numerical vectors using the CountVectorizer method. This produces a sparse representation using scipy.sparse.csr_matrix.

3) Filtering the data to remove words and sentences with '@', '#', links, repeated words, symbols, numbers, and lowercase words.

4) Lemmatization, which involves reducing words to their basic form by removing affixes and suffixes.

5) Removing stop words, which are general and less important words that appear frequently compared to other words.

2.3. Model

Following the preprocessing stage, the data is trained using three different algorithms: a deep learning model, Sequential Neural Network, and two traditional classification algorithms, Logistic Regression and Random Forest. These models will be discussed in further detail later. In this research, sequence analysis is employed to identify patterns within a series of events known as a sequence.

2.3.1 Sequential neural network preliminary knowledge

Sequence modeling is a machine learning technique that involves generating a sequence of values based on input data. This data can be time-series data, such as demand for a product over time, or text data, where the model predicts the next word based on a set of conditions and rules. Sequence modeling can be applied to various data types, including time series, DNA sequences, and meteorological data. In this particular study, sequence modeling is used to analyze the word sequence in a sentence and classify the sentiment expressed in the text. Sequential data refers to data in which the points are related to each other in some way, such as in a time series where each point represents an observation at a specific time. Sequence modeling can be used to identify patterns and make predictions based on this data.

2.3.2 Sequential neural network implementation

Keras sequential neural network model with two dense layers is leveraged in this work. For a fully connected network layer, is a layer where all neurons have connections to all the neurons in the next layer. This type of layer allows the model to learn more complex relationships between the input and output data.

The two dense layers used are Rectified Linear Unit and the Sigmoid. Three parameters specified in each layer are units, kernel_initializer and activation. The unit's parameter defines the size of the output space for the dense layer, the kernel_initializer parameter determines how the weights are initialized, and the activation parameter specifies the activation function to use for the output of the layer. These parameters can be adjusted to help improve the performance of a Keras model.

In the first layer, 50 units are used, kernel_initializer, and activation are set to be "uniform" and "Rectified Linear Unit (ReLU)" accordingly. The second layer has an output space size of 1 with also a "uniform" kernel initializer, however, activated by a "sigmoid" function.

The uniform kernel initializer in Keras is a kernel initializer that generates random values from a uniform distribution. This means that all weight matrixes are randomly initialized from a uniform distribution within the specified range. For example, if the uniform initializer is used with a range of [0, 1], all values in the weight matrix will be randomly initialized within the range [0, 1]. The uniform initializer is useful because it can help to prevent the weights from becoming too large or too small, which can cause problems during training, such as vanishing or exploding gradients. It can also help to improve the convergence of the model during training.

Two epochs are used in this model. During each epoch, the model iterates over the training set, making predictions and updating the weights to reduce the loss. After each epoch, the model's performance on the training and validation datasets is evaluated, and the model weights are updated to improve the model's performance.

The number of epochs to train the model for is an important hyperparameter that determines how long the model will train for. Generally, the more epochs the model trains for, the better it will perform on the training data, but at the cost of increased training time.

The activation function is a key component of neural networks that determines whether a particular neuron should be activated or not. This process helps the neural network learn complex patterns in the data and normalizes the output of each neuron to a range between 1 and 0 or between -1 and 1. In this study, activation functions include Sigmoid and ReLU.

1) Sigmoid. The sigmoid activation function, on the other hand, is a smooth, s-shaped function that maps the input values to a range of 0 to 1. This function is often used in classification tasks, where the output values represent the probabilities of different classes. The sigmoid function has the advantage of producing smooth outputs that can help the model to converge more quickly during training. However, the sigmoid function can suffer from vanishing gradients, where the gradients become very small and training slows down. Eq. 1 for the sigmoid function:

$$f(x) = \frac{1}{1+e^{-x}} \tag{1}$$

2) ReLU. The rectified linear unit (ReLU) activation function is a simple and effective one. Its output equals to the input if the input is positive and is zero otherwise. This function is easy to compute and can produce sparse outputs, which can be useful for certain types of neural networks. Additionally, ReLU avoids or resolves the issue of missing gradients. However, the ReLU function can sometimes result in "dying neurons", where the output is always zero, which can cause problems during training. Eq. 2 represents the formula for the ReLU function.

$$\sigma(x) = \begin{cases} (0, \ x), & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{2}$$

Optimization is a key component in the training of neural networks. Essentially, the learning process is an optimization problem where the goal is to achieve the condition to minimize the function f(x), subject to certain constraints on x. This value of x can be a single scalar or a vector of either continuous or discrete values. There are several different optimization algorithms that can be used to find this value of x, including Adam, AdaGrad, and RMSProp. These algorithms seek to find the optimal value of x by adjusting the parameters of the neural network through a process of iterative improvement. By optimizing the parameters of the neural network, we can improve its performance and accuracy on a given task.

2.3.3 Logistic Regression

Logistic regression is commonly used for classification, particularly binary classification. It is simple to implement and can handle large datasets efficiently. The algorithm uses a logistic function to predict the categorical probability, based on a labeled dataset. The logistic function maps input values to a range between 0 and 1 where the value represents the probabilities. The model then uses a threshold value to convert the probabilities into binary predictions.

It has several parameters that can be adjusted to improve the model's performance. In this study, the models are emplyed with a penalty of “l1”. The solver algorithm is set to be ‘liblinear’, and a max iteration of 10000 is used.

The penalty parameter specifies the type of regularization to use for the model. Regularization is a technique used to prevent overfitting. The penalty parameter can take one of two values: l1 or l2. The l1 penalize the the absolute value of the weights, while the l2 penalty adds a term proportional to the square of the weights.

The solver parameter specifies the algorithm to use to find the optimal values for the model weights. Different solver algorithms have different properties and may be better suited to different types of data and problems. Some common solvers include lbfgs, sag, and liblinear.

The max_iter parameter specifies the maximum number of iterations to use when training the model. This determines the maximum number of times the model will update the weights to reduce

the loss. A larger max_iter value can allow the model to find a better solution, but will increase the training time.

2.3.4 Random forest

A random forest (RF) classifier combines the predictions of several decision trees to make a final prediction. It is an improvement on a basic decision tree classifier because it is more resistant to noise in the training data and is less prone to overfitting. RF classifiers are trained by training several decision trees in parallel using bootstrapping, which means that each tree learns from a different random subset of the training data. This helps to reduce the overall variance of the RF classifier. After training, the RF classifier makes predictions by averaging the predictions of all the trees. RF classifiers are accurate, robust, and can be used for both regression and classification tasks. They are also able to handle a large number of features and are not sensitive to data scaling. However, they can be harder to interpret than a single decision tree and require more computational resources to train.

"RandomForestClassifier" is an ensemble model in the sklearn library for classification tasks It has several parameters that can be adjusted to improve the model's performance. In this analysis, 200 estimators are used during the training process, and a random state of 42 is applied to this random forest model.

The n_estimators parameter specifies the number of decision trees to use in the ensemble. This is an integer value that determines the size of the ensemble. A larger number of trees can produce more accurate predictions, but will also increase the training time.

The random_state parameter specifies the random seed to use for the decision trees. This is a numerical value that determines the randomness in the training of the decision trees. Setting the random_state to a fixed value can make the results of the model reproducible, but using different random seeds can produce different ensembles of trees, which can improve the model's performance.

2.4. Evaluation matrix

The confusion matrix in Figure 3 is a visual representation of how well a classifier is performing. It shows the number of true positive (TP) values, denotes the correctly classified positive samples, as well as false positive (FP) values, which are examples that are incorrectly classified as positive samples. It also shows false negative (FN) values, which are examples that are incorrectly classified as negative class, and true negative (TN) values, which are examples that are correctly classified negative class. The confusion matrix is used to calculate various performance metrics, such as accuracy (ACC), precision (P), sensitivity (Sn), specificity (Sp), and F-score. These metrics provide insights into how well the classifier is performing and help to identify areas for improvement.

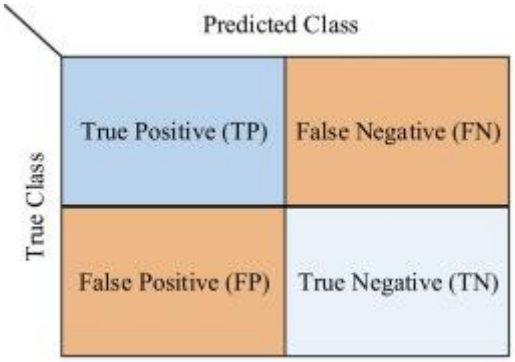

Fig. 3 Example of confusion matrix.

3. Result and discussion

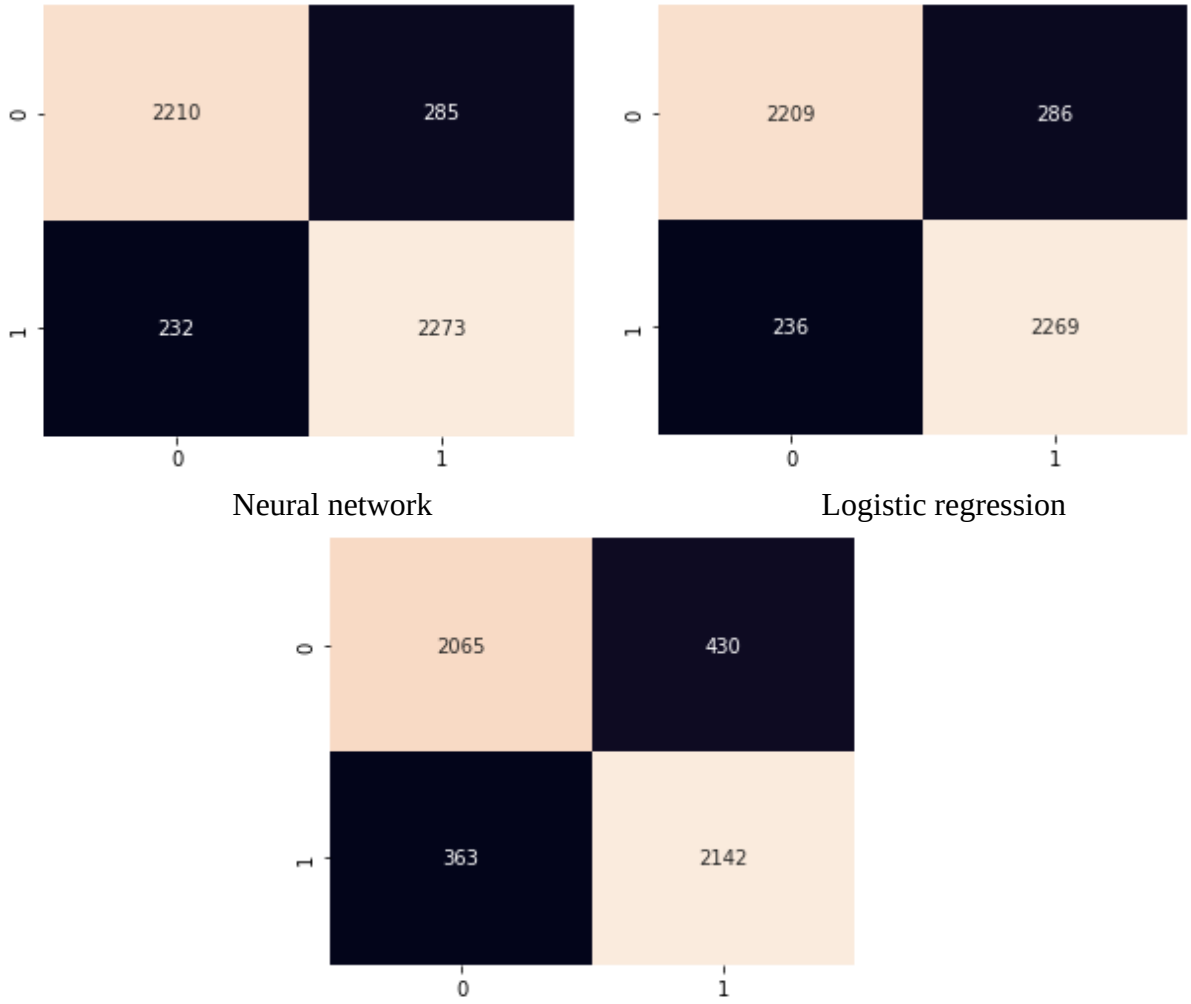

Fig. 4 Confusion matrix results.

Figure 4 demonstrates the performances of the confusion matrix. Table 1 shows the accuracy, precision, and recall rates for three different machine learning algorithms: sequential neural network, logistic regression, and random forest.

The sequential neural network achieved an accuracy of 90%, which indicates that it correctly predicted the sentiment of tweets in 90% of cases. This is a relatively high accuracy, indicating that the model is able to accurately classify tweets as positive or negative. The precision of the sequential neural network was 89%, which means that out of all the tweets that the model predicted as positive or negative, 89% of them were actually positive or negative. This is a relatively high precision, indicating that the model is not making many false predictions. The recall of the sequential neural network was 90%, which means that out of all the tweets that were actually positive or negative, the model was able to correctly identify 90% of them. This is a relatively high recall, indicating that the model is able to capture a large proportion of the actual positive and negative tweets.

The logistic regression algorithm has a similar performance to the sequential neural network, with a 90%accuracy, a 89.5% precision, and a 90% recall. This suggests that the logistic regression model is slightly more precise than the sequential neural network, but has the same accuracy and recall rate.

The random forest algorithm has a lower accuracy rate of 83%, a 83% precision, and a 84% recall. This indicates that the random forest model is less accurate and precise than the other two algorithms, but has a slightly higher recall rate.

Table 1. Performances of three algorithms.

	Accuracy (%)	Precision (%)	Recall (%)
Sequential neural network	90	89	90
Logistic regression	90	89.5	90
Random forest	83	83	84

In comparison, the logistic regression model also achieved a high accuracy and precision, but with slightly lower recall than the sequential neural network. This indicates that the logistic regression model is slightly less effective at capturing all the actual positive and negative tweets.

The random forest model, on the other hand, had lower performances than the other two models. This indicates that it is less effective at predicting the sentiment of tweets and may be making more false predictions.

To sum up, the sequential neural network appears to be the most effective method for sentiment analysis, followed by logistic regression, with random forest being the least effective.

Figure 5 and Figure 6 are depicted to show some further findings in terms of the sequential neural network and random forest. This provides more concrete evaluation of the comparison of the three algorithms.

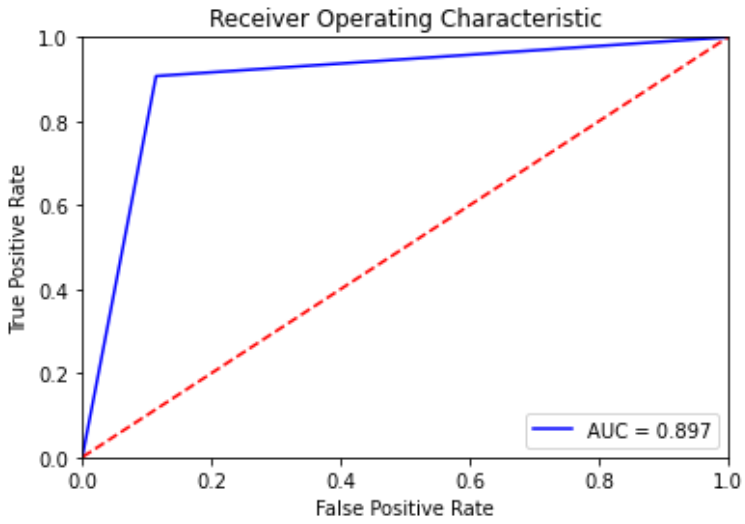

Fig. 5 AUC result of the sequential neural network.

In a classification task, the threshold is a value that is used to convert the predicted probabilities into binary predictions. If the predicted probability is greater than the threshold, the example is predicted to belong to the positive class, otherwise, it is predicted to belong to the negative class.

The receiver operating characteristic (ROC) curve is a graphical representation of a classifier's performance, where the y-axis is true positive rate (TPR) and the x-axis is the false positive rate (FPR). The ROC curve shows how the TPR and FPR change as the threshold value is varied, and how they trade off against each other. The area under the ROC curve (AUC) is a metric that measures the overall performance of a classifier at different threshold values. AUC ranges from 0 to 1, the higher the better. An AUC of 0.5 represents a random classifier, while a perfect classifier is indicated by an AUC of 1.0. In this case, the classifier achieved an AUC of 0.897, indicating that it performs well across a range of threshold values and has a strong ability to differentiate between positive and negative tweets.

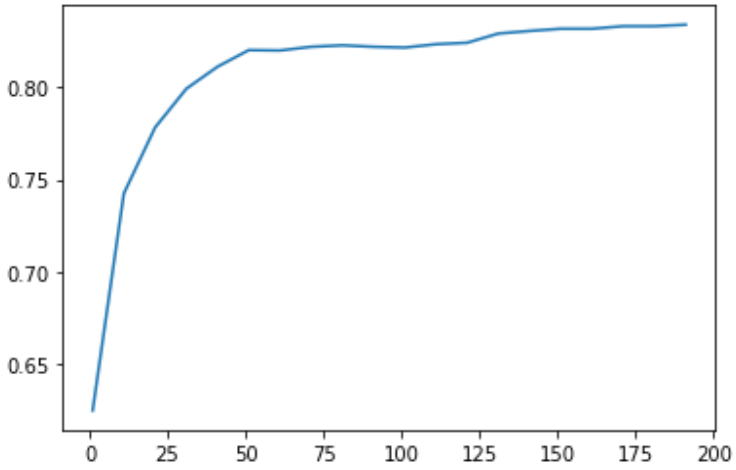

Fig. 6 Relationship between the number of estimators and accuracy in RF.

The graph shows the relationship between the number of estimators in a random forest algorithm and the accuracy of sentiment analysis. As the number of estimators increases, the accuracy of sentiment analysis also appears to increase. This suggests that using more estimators in a random forest algorithm can improve the accuracy of sentiment analysis. This may be because using more estimators allows the algorithm to consider a larger number of decision trees, which can enhance the overall accuracy of the model. However, it's worth noting that increasing the number of estimators may come at the expense of increased computational time and resources. Therefore, it's important to find a balance between the number of estimators and the accuracy of sentiment analysis.

4. Conclusion

The experiment conducted shows the performance of various machine learning models in the context of sentiment analysis task about the IMDb Movie reviews. The sequential neural network and logistic regression have similar accuracy levels of 90%, while the random forest has a slightly lower accuracy of 83%. The sequential neural network used in this study has two dense layers, activated by relu and softmax functions, which are common activation functions in neural networks. The logistic regression model used in this study has an l1 penalty, which is a type of regularization technique that can help prevent overfitting. The random forest model used in this study has 100 estimators. In the context of IMDb Movie reviews, sentiment analysis can be used to gain insights into the overall sentiment of a movie and how it is perceived by the public. This information can be useful for movie studios and producers, as it can help them gauge the success of their films and make decisions about future productions. Overall, these algorithms have demonstrated strong performance in the context of sentiment analysis of IMDb Movie reviews.

References

[1] Feldman R. Techniques and applications for sentiment analysis. Communications of the ACM. 2013, 56(4):82-89.

[2] Whitelaw C, Garg N, Argamon S. Using appraisal groups for sentiment analysis. In Proceedings of the 14th ACM international conference on Information and knowledge management 2005, 625-631.

[3] Ortigosa A, Martín JM, Carro RM. Sentiment analysis in Facebook and its application to e-learning. Computers in human behavior. 2014, 31:527-41.

[4] Jain AP, Dandannavar P. Application of machine learning techniques to sentiment analysis. In 2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT) 2016, 628-632.

[5] Smailović J, Grčar M, Lavrač N, Žnidaršič M. Predictive sentiment analysis of tweets: A stock market application. InInternational workshop on human-computer interaction and knowledge discovery in complex, unstructured, big data 2013, 77-88.

[6] Yenter A, Verma A. Deep CNN-LSTM with combined kernels from multiple branches for IMDb review sentiment analysis. In2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON) 2017, 540-546.

[7] García-Cumbreras MÁ, Montejo-Ráez A, Díaz-Galiano MC. Pessimists and optimists: Improving collaborative filtering through sentiment analysis. Expert Systems with Applications. 2013, 40(17):6758-65.

[8] Kumar HM, Harish BS, Darshan HK. Sentiment Analysis on IMDb Movie Reviews Using Hybrid Feature Extraction Method. International Journal of Interactive Multimedia & Artificial Intelligence. 2019, 5(5):109-114.

[9] Hudlicka E. To feel or not to feel: The role of affect in human–computer interaction. International journal of human-computer studies. 2003, 59(1-2):1-32.

[10] Lakshmipathi. IMDB Dataset of 50K Movie Reviews. 2018. URL: https://www.kaggle.com/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews