SENTIMENT ANALYSIS ON MOVIE REVIEWS

TASK 1

Student Name:	James Antony Das
Matriculation Number:	9216076
Course Name:	Project: NLP
Tutor Name:	Dr. Simon Martin
Submission Date:	31 January 2025
University:	IU, International University of Applied Sciences

Abstract

Sentiment analysis of movie reviews is a crucial task for assessing the public opinions of the movie. In this project I have developed a natural language processing system that can analyse movie reviews and classify them as positive or negative. This project report explores the steps and decisions of building such system. Two datasets were chosen, one for training the model and another for testing the model. First the given dataset was cleaned, stop words were removed, and word lemmatization was applied. Several word and sentence embedding techniques were tried and TF-IDF is chosen as the final embedding. The generated embeddings were then scaled. Principle component analysis (PCA) was used for dimensionality reduction of the generated embeddings. Finally, several machine learning techniques including deep neural networks were tried and Support Vector Machine (SVM) Classifier was chosen based on the appropriate performance metrics. The developed model is then tested on the bigger test dataset, and the testing performance were presented in this report.

The project Jupiter notebook is available on the GitHub repository:
https://github.com/jamesantonydas/Sentiment_Analysis_movie_reviews

Table of Contents

List of Figures

List of Tables

List of Abbreviations

DNN – Deep Neural Network

GLOVE – Global Vectors for Word Representation

HTML - Hypertext Markup Language

PCA – Principal Component Analysis

SVM – Support Vector Machine

TF-IDF – Term Frequency Inverse Document Frequency

TP – True positive

FP – False positive

TN – True Negative

FN – False Negative

Word2Vec – Word to Vector

1. Introduction

Assessing sentiment of movie reviews could be a challenging task for websites, as they must go through a wall of text for understanding the opinions of the reviewer. These reviews could be positive, or negative, sometimes neutral as well (Alzate et al, 2024). To observe the overall user opinion of the movie, it is important to accurately understand the sentiment of movie reviews.

The goal of this project is to develop a machine learning algorithm for assessing the sentiment of the given movie review and classify them as positive or negative. The classification model is built on a labelled dataset of movie review. The intent was to build a system that is efficient, accurate and computationally inexpensive to handle huge dataset. Several methodologies have been tried and tested, and the best performing methodology was selected for building the final model.

First section of this project report explores the datasets used for training and testing, the overall architecture of the developed model, and the performance metrics used for evaluating the model. The later sections explore the individual steps, such as data cleaning, word lemmatization, word embeddings with Term-frequency and Inverse Document Frequency (TF-IDF), Feature Scaling with Min max scaler, Dimensionality reduction with Principal component Analysis (PCA), and Machine learning with Support vector machine (SVM) classifier.

Later sections explain the decision in designing the architecture of the model, such as insights gained from visualizing the word embeddings in 3D with principal components, and estimating the appropriate cumulative variance explained for dimensionality reduction. Finally, the performance of the developed system has been tested on a huge dataset of labelled movie reviews, and the results were presented in this report.

2. Related works

For better understanding of the concepts, it is important to understand the projects that are related to the scope of this project. Hermawan et al, (2024) have developed an Enhanced sentiment analysis algorithm using Support vector machine algorithm. Their project tries to quantify the presence of emotions such as anger, joy, surprise, present in the movie review.

Danyal et al., (2023) has developed a sentiment analysis model using TF-IDF, and count vectorizer. They have utilized Naive Bayes (NB) Classifier as the supervised machine learning. For better accuracy of their system, they have utilized effective Data cleaning techniques, such as spell checks, and fixing chat words. They have utilized hyperparameter tuning for improving the accuracy of their model.

3. Overview of the System

Since sentiment analysis is a classification task, we need a huge dataset of movie reviews which are labelled as positive or negative based on the sentiment for training a Supervised learning model. Selecting a labelled dataset is a vital part of training the machine learning model. This section explores the datasets that were used for training and testing the system. And then, the overall architecture of the system. Finally, explains the performance metrics which will be used to evaluate the system.

3.1 Datasets

For this project, two different datasets were selected, one for training the machine learning model, and one for evaluating the performance and generalization of the system.

Polarity v2.0 dataset by Pang (2004) was used to train the system, the dataset has 2000 entries, 1000 of them are labelled as positive, and 1000 of them are labelled as negative (Pang and Lillian Lee, 2004). The dataset is balanced.

The training dataset has been divided into training and testing datasets using sklearn's *train_test_split* function, 90% of the dataset has been allocated for training, and 10% of the dataset as been allocated for initial testing, in this case, Validation of the model.

Exploratory data analysis was conducted on the training dataset. Word Cloud plot reveals the most frequently used words in the corpus such as *film, movie, character,* etc. Machine learning algorithms are good at classifying such data.

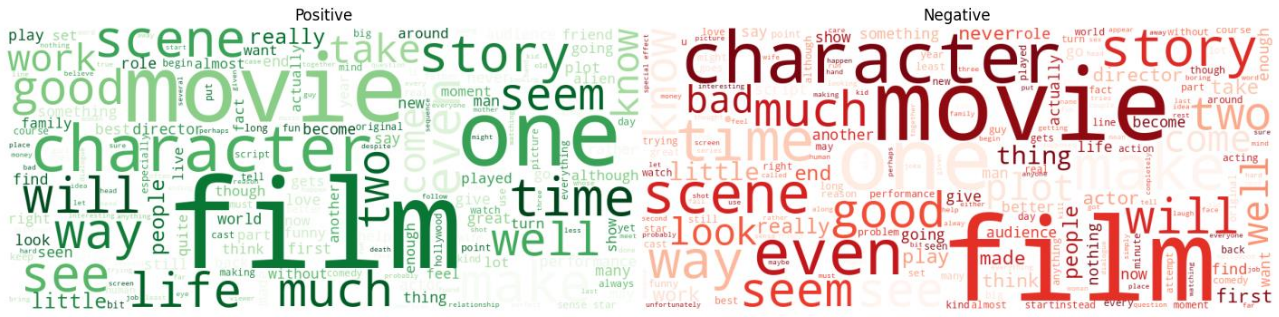


Figure 1

Large Movie review Dataset by Maas et al. (2011), was used to test the developed system, the dataset has 50,000 entries, 25,000 of them are positive and 25,000 of them are negative reviews (Maas et al., 2011). Like the training dataset, the test dataset is balanced as well.

3.2 Model Pipeline

The given training data was cleaned, and pre-processed. The stop words were removed from the text and lemmatization process was conducted. The text was vectorized with Word embedding techniques. Several word embedding techniques were compared, and the resulting vectors were visualized in 3D using principal component analysis (Power, 2022). Term-frequency and inverse document frequency (TF-IDF) as the word embedding technique for this project because TD-IDF embeddings were linearly separable.

The generated word embeddings were scaled with Minmax Scaler which brings the features values between 0 and 1. This would bring all the features to the same scale. Dimensionality reduction was conducted using Principal component analysis. Approximately 90% cumulative variance explained was calculated, and appropriate number of principle components were chosen.

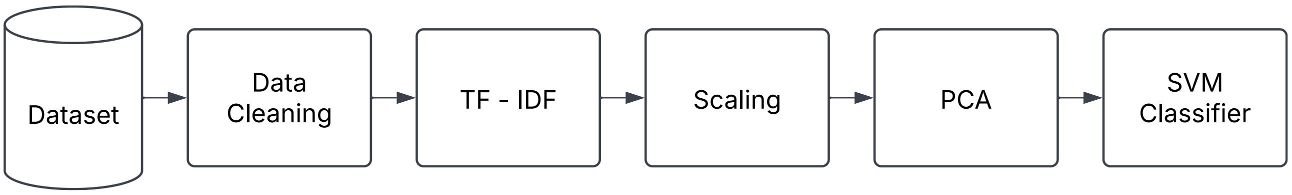


Figure 2

Finally, A machine learning model was trained on the generated embeddings with label of the dataset. The best model was chosen based on the performance on the validation set. The final model was tested on the large dataset. These steps were explained in great details in the upcoming sections.

3.3 Performance metrics

Since the sentiment analysis is a classification problem, we will evaluate the performance of the model with four different metrics, Accuracy, precision, recall and F-score.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{TN}}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 - score = 2 * \frac{precision * recall}{precision + recall}$$

Where, TP, TN, FP, FN represents True positive, True Negatives, False positives, and False Negatives respectively.

While performing hyperparameter tuning, it has been decided to focus on improving the Accuracy of the machine learning model, since the given dataset is balanced. Our focus will be to build a model that is neither overfit nor underfit on the given dataset.

4. Data Preprocessing

4.1 Data Cleaning

The given training dataset was cleaned using Regular expressions. First, the entire corpus was converted to lower case, that would make the words consistent. Then punctuations, special characters, HTML tags, numbers, usernames and unnecessary spaces were removed from the given dataset. This cleans up the dataset. Secondly, a function to remove the stop words was applied to the text. The stop words are the words that doesn't have meaning. While stop words provide the vital context to the given text, the stop words offer no significant contributions for the statistical analysis for the text (Chaerul Haviana et al., 2023)

Before Data Cleaning	After Data Cleaning and Stop words removed
plot : set in the future , a courier has uploaded..	plot set future courier uploaded …
one night , during a torrential downpour that..	one night torrential downpour flooded …

Table 1

4.2 Word Lemmatization

In Natural language processing, the text is converted to standardized form by stemming or converted to their root word through Lemmatization (Kadhim, 2018). In this project, after removing the stop words, we apply word Lemmatization to the corpus. The effect of Lemmatization to the reviews is presented below. It is also important to note that removal of stop words and application of Lemmatization may, in some cases, result in damage of context of the text (El Kah and Zeroualet, 2021). In our case, use of word Lemmatization increases the performance of the machine learning algorithm.

Before Lemmatization	After Lemmatization
plot set future courier uploaded …	plot set future courier uploaded …
ads make hanging seem like upbeat comedy…	ad make hanging seem like upbeat comedy ...

Table 2

4.3 Word Embeddings

Word embeddings is a popular way of converting the given text to vectors that can be processed by the machine learning algorithms, while preserving the semantic relationship among words (Tbaikhi et al, 2024).

Several Embedding techniques were tried including Word2Vec, GLOVE, Term frequency Inverse Document Frequency (TF-IDF), and Sentence embedding using Sentence Transformer. These embedding techniques generates an embedding vector of given size from the text format. The resulting embeddings were visualized in 3d using Principal components analysis (PCA).

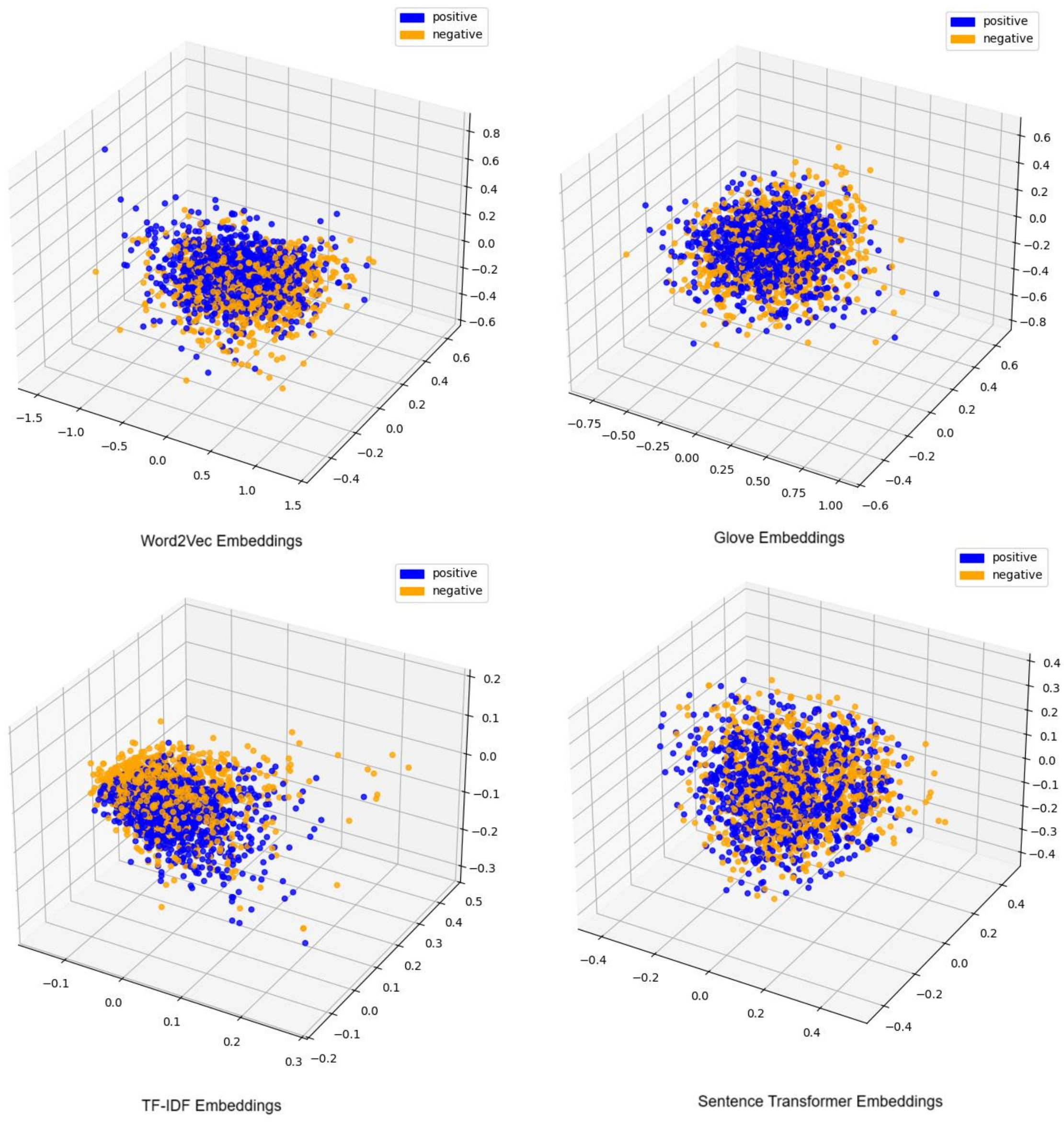


Figure 3

Visualizing the different embeddings, along with their labels, reveals a distinct pattern, such that the similar datapoints were closer to each other. And in the case of TF-IDF, the positive and negative reviews are seemingly linearly separable. Machine Learning algorithms such as SVM Classifier tends to perform best when the data is linearly separable. TF-IDF is mathematically given by,

$$W_{X,Y} = TF_{X,Y} * log(\frac{N}{DF_X})$$

Where TF is the term frequency, DF is the document frequency of X, and N is the total number of documents (Sadak, 2023). The TF-IDF embeddings was picked because they were visually linearly separable, when visualized in 3d. Since TF-IDF generates huge sparse matrix, it would be best to reduce the dimensions of the embedding vector with principal components.

4.4 Feature Scaling

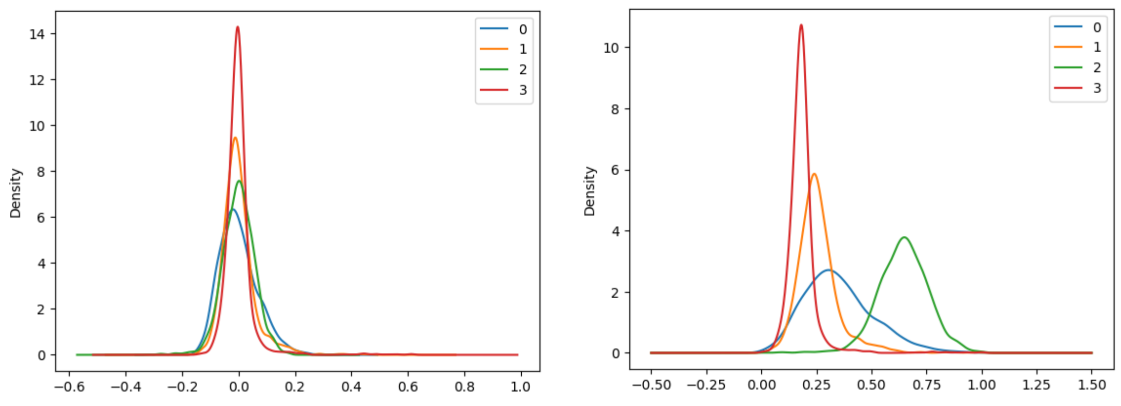
Data Transformation with scaling is an important step for improving the data quality while building the machine learning technique. Standard scaler and Min max scaler are the popular scaling techniques for scaling the data before machine learning (Munkhdalai et al., 2019).

Our experiments with feature scaling showed that the min max scaling outperformed Standard scaling for our project by 10%. Therefore, for this project, Min max scaling would be used for normalizing the generated embedding. According to Scikit-learn's documentation (2025), the min max scaling is mathematically expressed as,

$$X_{std} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

$$X_{scaled} = X_{std} * (max - min) + min$$

Where min, max are the feature range of the given scaler. For example, between zero and one. The effects of feature scaling are detailed in the following image.



Before Scaling (4 principal components) After Min Max Scaling

Figure 4

4.5 Dimensionality Reduction with PCA

Dimensionality reduction techniques such as principal component analysis (PCA) not only decrease the computational expenses of the model. But also decreases the noise in the data. Application of PCA removes the tendency of a model to overfit on the training set. This helps the model learn the more general trend from huge sparse dataset (Yifan Xie et al, 2024).

The number of components needed to explain the variance is estimated for each word embeddings and a number is chosen for building the model. In the case of TF-IDF, it is estimated that approximately, 150 components were enough to represent the embedding data. By applying PCA, we have decreased the features of TF-IDF embeddings from 42,000 to 150. Thereby increasing the efficiency and generalization of the model (Youngkeun Choi, 2024).

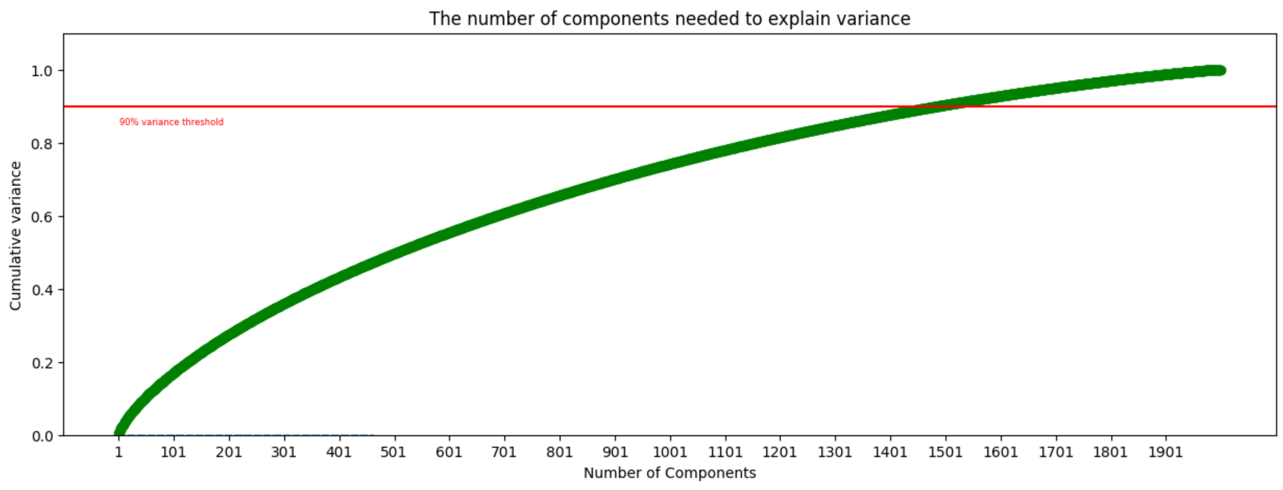


Figure 5

5. Machine Learning

The final step is to build a machine learning model on top of the processed data. The training dataset is processed and used for building the machine learning pipeline. Second, Hyperparameter tuning was performed for improving the score of the machine learning model (Cassia, 2023).

5.1 Model Building

Several machine learning algorithms including Support Vector machine (SVM) classifier, Random Forest classifier, and Deep Neural Networks has been trained, and tested on the initial validation datasets. The results of the model on validation set are presented below.

It is important to notice that random forest classifier was quick to overfit on the given dataset when training, within 10 epochs, even after feature reduction. This makes the random forest classifier unfit for the given task. Deep Neural networks are performing remarkably well but not as good as SVM. most probable explanation is that the given number of data entries (2000 rows) is not enough for effectively train the deep neural network of this scale (Baheti, 2021), while SVM is good at tasks that are linearly separable.

Model	Embedding	Accuracy	Precision	Recall	F1score	Comment
SVM Classifier	Word2Vec	0.61	0.62	0.55	0.59	
	Glove	0.73	0.74	0.70	0.72	
	TF-IDF	0.87	0.87	0.87	0.87	Best model
	Sentence Transformer	0.69	0.73	0.61	0.67	
Random Forest Classifier	Word2Vec	0.63	0.64	0.62	0.63	
	Glove	0.70	0.71	0.69	0.70	
	TF-IDF	0.75	0.73	0.81	0.77	
	Sentence Transformer	0.67	0.69	0.63	0.66	
Deep Neural Network	Word2Vec	0.71	0.73	0.66	0.69	
	Glove	0.68	0.72	0.59	0.65	
	TF-IDF	0.81	0.83	0.78	0.80	
	Sentence Transformer	0.65	0.68	0.59	0.63	

Table 3

Among all the models that were tried on various embeddings, SVM offers a balanced performance on the base model.

5.2 Hyperparameter Tuning

Hyperparameter tuning was conducted using Grid Search and cross-validation function from Scikit learn library. A cross validation of 5 folds were conducted during the Hyperparameter tuning. While Hyperparameter tuning didn't not improve performance of the models across various metrics. It is important to notice that tuned model is less overfit on the training data compared to the base model.

Model	Dataset	Accuracy	Precision	Recall	F1 Score
Base SVM	Training	0.95	0.9513	0.9536	0.9525
	Validation	0.8625	0.8613	0.8656	0.8635
Tuned SVM	Training	0.925	0.928	0.922	0.925
	Validation	0.8625	0.861	0.865	0.863

Table 4

6. Results and Recommendations

6.1 Model Evaluation on the Test dataset

The developed model is tested on the Large Movie review dataset, which has 50,000 entries (Maas et al. 2011). The model is performing well on the large dataset with 81% Accuracy. The performance of the model is presented in the table below.

	Accuracy	Precision	Recall	F1 Score
Training dataset	0.92	0.928	0.922	0.925
Test dataset	0.81	0.82	0.797	0.811

Table 5

The confusion matrix of the training and testing dataset is presented below. The developed model is neither overfit nor underfit on the training dataset. And more importantly, the model is generalized well. And the final model is performing efficiently with minimal computational expense. Finally, the model can be used to assess the overall sentiments of individual movies.

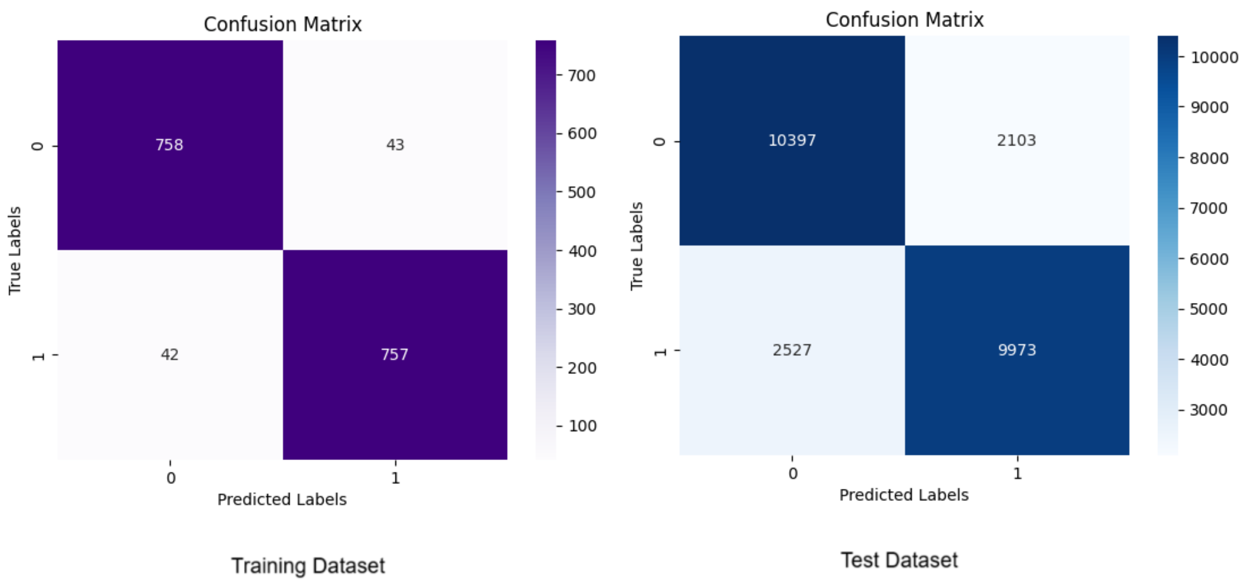


Figure 6

6.2 Business Recommendations

While the developed model can be utilized for accurately classifying movie reviews, it is important to try more computationally expensive models such as Transformers or Large language models, which may be aware of the context of the models.

Also, it is beneficial to try training Deep learning models such as Long short-term memory (LSTM), Recurrent neural networks (RNN), etc. with vast amounts of training data to avoid overfitting of the model. These steps could improve the accuracy and performance of the model. Furthermore, in terms of machine learning, Ensemble models that are based on different embeddings can also be experimented for better performance of the model.

The python notebook of the project is available in the GitHub repo:
https://github.com/jamesantonydas/Sentiment_Analysis_movie_reviews

7. Conclusion

In this project, we have developed a sentiment analysis system for identifying the movie reviews as positive or negative. The developed model can process huge volume of movie reviews and classify them, accurately and efficiently with minimal resources.

We have picked two different datasets, one for Training and a larger one for testing. An extensive Exploratory Data Analysis was performed using word cloud plots. Both the training and testing datasets are balanced. 10% of the Training dataset was allocated for validating the models. The goal was to pick the base model that was not overfit on the training dataset.

Through Data cleaning was performed using regular expressions. HTML tags, usernames, numbers, and punctuations were removed. Stop words were removed from the text. And Word Lemmatization was applied to the training dataset. Term frequency Inverse document frequency (TF-IDF) was used for generating word embeddings, the embedding vectors was scaled using Min max scaler and reduced to 150 dimensions using principal component analysis (PCA).

Finally, Support Vector Classifier (SVM) was trained on top of the embeddings, using the labels as target variables. Hyperparameter tuning was performed for increasing the accuracy of the model. While the Hyperparameter tuning didn't improve the accuracy of the model, the tuned model was less overfit on the training dataset. Finally, the model is tested on the large movie review dataset. And the performance and generalization of the model is evaluated using performance metrics and confusion matrix. The developed model is 81% accurate on the testing dataset.

The accuracy of the sentiment analysis system can be improved by trying more complex, context aware deep learning models, such as Transformers and Large language models. The final model can be deployed for classification of movie reviews.

References

Alzate, M., Arce-Urriza, M., & Cebollada, J. (2022). Mining the text of online consumer reviews to analyze brand image and brand positioning. *Journal of Retailing and Consumer Services, 67*, 102989. https://doi.org/10.1016/j.jretconser.2022.102989

Baheti, P. (2021, December 1). What is overfitting in deep learning [+10 ways to avoid it]. *V7 Labs.* Retrieved from https://www.v7labs.com/blog/overfitting

Chaerul Haviana, S. F., Mulyono, S., & Badie'Ah. (2023). The Effects of Stopwords, Stemming, and Lemmatization on Pre-trained Language Models for Text Classification: A Technical Study. *2023 10th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), Electrical Engineering, Computer Science and Informatics (EECSI), 2023 10th International Conference On*, 521–527. https://doi.org/10.1109/EECSI59885.2023.10295797

Danyal, M. M., Khan, S. S., Khan, M., Ullah, S., Ghaffar, M. B., & Khan, W. (2024). Sentiment analysis of movie reviews based on NB approaches using TF–IDF and count vectorizer. Social Network Analysis and Mining, 14(1). https://doi.org/10.1007/s13278-024-01250-9

El Kah, A., & Zeroual, I. (2021). The effects of pre-processing techniques on Arabic text classification. *International Journal, 10*(1), 1–12.

Hermawan, A., Yusuf, R., Daniawan, B., & Junaedi. (2025). Enhanced sentiment analysis and emotion detection in movie reviews using support vector machine algorithm. *Telkomnika*, *23*(1), 138–146. https://doi.org/10.12928/TELKOMNIKA.v23i1.26377

Kadhim, A. I. (2018). An evaluation of preprocessing techniques for text classification. *International Journal of Computer Science and Information Security, 16*(6), 22–32.

Munkhdalai, L., Munkhdalai, T., Park, K.H., Lee, H.G., Li, M., Ryu, K.H. Mixture of activation functions with extended min-max normalization for forex market prediction. IEEE Access 7, 183680–183691 (2019) https://doi.org/10.1109/ACCESS.2019.2959789

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011). Learning word vectors for sentiment analysis. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 142–150. Association for Computational Linguistics. http://www.aclweb.org/anthology/P11-1015

Pang, B., & Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. *Proceedings of the ACL.* https://www.cs.cornell.edu/people/pabo/movie-review-data/

Power, B., & Sanders, T. (2022, March 10). Visualizing embeddings in 3D. *OpenAI Cookbook.* Retrieved from https://cookbook.openai.com/examples/visualizing_embeddings_in_3d

Sadak, V. (2023, March 26). TF-IDF from scratch with Python. *Medium.* Retrieved from https://medium.com/@rebirth4vali/tf-idf-from-scratch-with-python-e22033bb99f

Sampaio, C. (2023, April 21). Understanding SVM hyperparameters. *Stack Abuse.* Retrieved from https://stackabuse.com/understanding-svm-hyperparameters/

Scikit-learn developers. (n.d.). *MinMaxScaler*. Scikit-learn documentation. Retrieved January 31, 2025, from https://scikit-learn.org/stable/modules/preprocessing.html#minmaxscaler

Tbaikhi, S., Jakha, H., ElHoussaini, S., ElHoussaini, M.-A., & ElKafi, J. (2024). New Approach Based on Word Embedding and Deep Learning Algorithms to Optimize the Sentiment Analysis Performance in Social Business Intelligence. *2024 Sixth International Conference on Intelligent Computing in Data Sciences (ICDS), Intelligent Computing in Data Sciences (ICDS), 2024 Sixth International Conference On*, 1–7. https://doi.org/10.1109/ICDS62089.2024.10756441

Xie, Y., Wang, T., Kim, J., Lee, K., & Jeong, M. (2024). Least angle sparse principal component analysis for ultrahigh dimensional data. *Annals of Operations Research*, 1–27. https://doi.org/10.1007/s10479-024-06428-0

Youngkeun Choi. (2024). Startup Success Prediction with PCA-Enhanced Machine Learning Models. *Journal of Technology Management & Innovation*, *19*(4), 77–88.