Project Write-Up: Performing Sentiment Analysis on Movie Reviews
Name: Maha Attique, Andric Brena
Course: CS360 Machine Learning
Instructors: Sara Mathieson, Sorelle Friedler

For any product, customers may seek to read reviews to gain insight on different perspectives regarding the product. This is to help facilitate their decision making on whether to engage with the product as they learn how someone else perceived it. Movies are one form of entertainment subject to numerous reviews, and to aid in understanding how a reviewer feels about a movie, sentiment analysis can be done. This form of analysis is a process which seeks to classify text as having a sentiment such as ‘positive’ and ‘negative’ or ‘joy’ and ‘sadness’. As such, it can be used to determine the attitude a reviewer had and to disclose the general trend or feeling people have towards a movie. This serves as one area of interest we have for analyzing movie reviews. Another thing to consider is that reviewers and movie goers in general are seen as the customers, those who are supplied a product by a producer. In other words, the vision or sentiment the producer may have towards their movie may not be the same as the feeling viewers gain from watching the movie. Given an official description of the movie from the creators, we found it to be an interesting avenue to explore whether a review elicited the same feeling as that found in these descriptions. Thus, we asked ourselves how can we design a model that is able to take in a movie review as an input and produce two outputs, one detailing its ‘positive’ or ‘negative’ sentiment in general and the other detailing the emotion such as ‘joy’ or ‘anticipation’ with respect to the list of possible emotions found in a set of movies. To realize this goal, we decided for this project to break it down into two models to analyze different aspects of the set of movie reviews separately but to use the results in conjunction to gain an overall understanding on the spread of positive and negative reviews and how well they elicited the same emotion found in their respective movie descriptions.

Our data is the IMDB Movie Reviews dataset of size n=50000 provided through Kaggle and originally constructed in “Learning Word Vectors for Sentiment Analysis” (Mass et al., 2011). It originally contained the features: ratings, reviews, movie names, and resenhas(translation) before it was extended to also contain genres, descriptions, and emotions. Ratings were int values on a scale from 1-10 reflecting the scores critics had for a movie while the rest are text data. Specifically, movie names referred to the name of the movie, genres the categories a movie was placed in, reviews critical analysis based on thoughts and observations, ‘resenhas’ translations of the reviews to a different language, descriptions written analysis from a distant point of view, and emotions the feelings emitted from a description. One important thing to note is that movies tended to have more than 1 description meaning that they also had more than 1 emotion, and rather than pairing these descriptions or emotions together in a list, they were separated. In other words, there were rows with duplicated reviews such that one row example had a review with one emotion and corresponding description while another row had the same review but with the other emotion and its corresponding description. Rather than combining the emotions for duplicated reviews, we decided to drop the duplicates such that each row had a unique review with only 1 emotion attached to it rather than 2 or more. This considerably reduced the dataset size to 19316 as this is the number of unique reviews.

Our goal is to create two models: a SVM and an RNN. SVM will focus on classifying reviews with a ‘positive’ or ‘negative’ sentiment. These binary sentiments are derived from a row’s rating: if the rating is > 5, the review is labeled as ‘positive’ or 1, and if the rating is <= 5, the review is labeled as ‘negative’ or -1. RNN will focus on classifying reviews with an emotion such as ‘joy’ or ‘sadness’ from the provided descriptions column. In general, our main feature is the reviews column for both models where the binary sentiment serves as the class for SVM and emotions as the class for RNN.

Before we could apply these algorithms, we had to prepare our data. Our data was downloaded as a csv file, so we used pandas to read it in as a dataframe. First, we dropped rows with duplicated reviews and then shuffled the remaining rows. We decided to build subsets of different sizes such as 2500 and 10000. For each subset, we added a rating_labels column pertaining to a review’s binary sentiment. We also took into account the distribution of emotions, ratings, and binary sentiment so that we can perform preliminary analysis to inform our models’ results. Finally, because our main feature is the reviews column, we filtered each review. Each review was tokenized or broken down into individual words and sent through a process which eliminates stopwords or words common in the English language that generally have no feeling and reduces the tokens to their base form based on its part of speech using lemmatization. These tokens were then combined back together to rebuild the now filtered review. Most importantly, because the reviews are text data, we decided to vectorize each review into numerical values using the bag of words method where each word gets attached a value representing its count. After this step, the data is now prepared to get sent through our models.

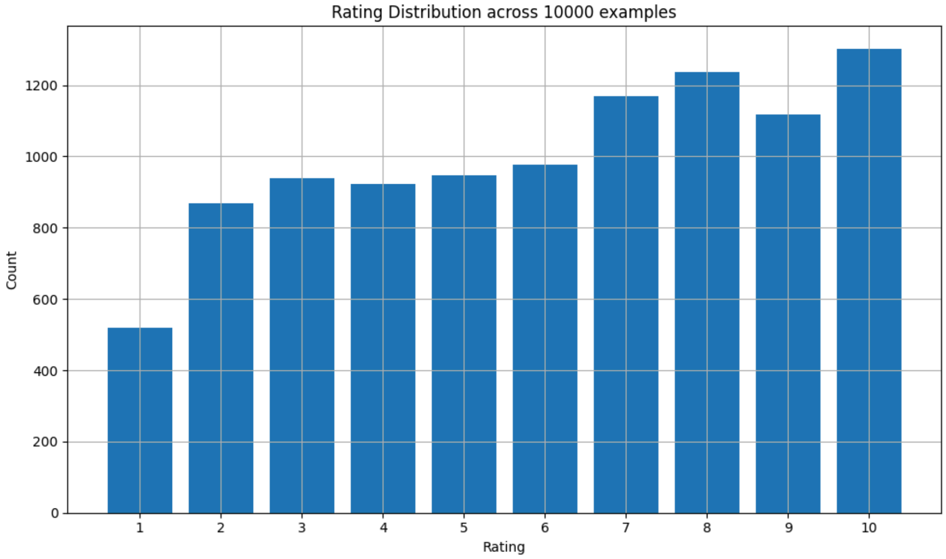


Figure 1: The occurrence of each rating across the data subset of size 10000.

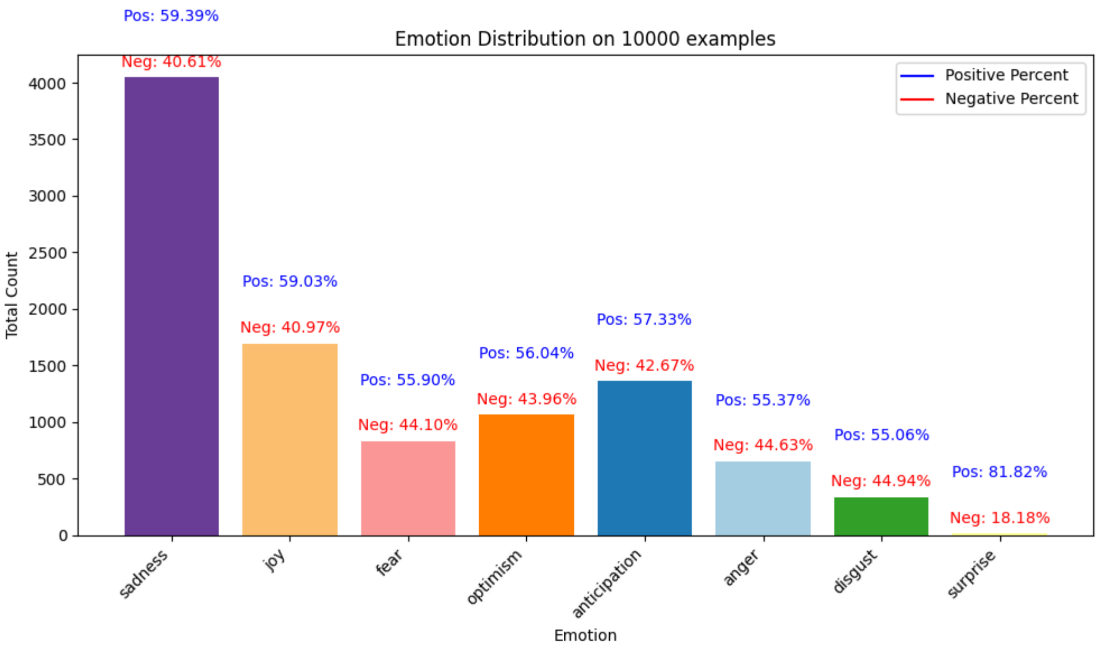
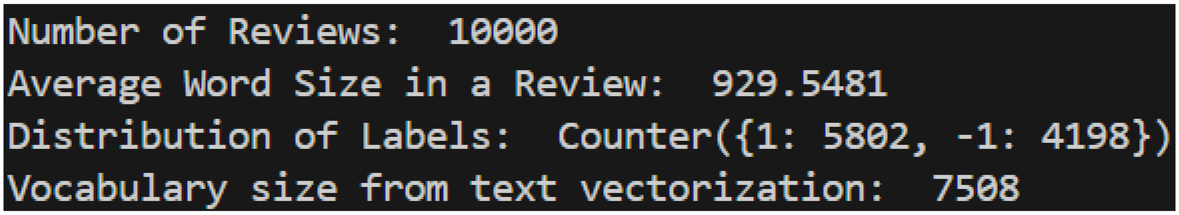


Figure 2: The distribution of emotions and spread of binary sentiments for each one across the data subset of size 10000.

Before running the model algorithms, we sought to perform preliminary analysis on our preprocessed datasets. Here, we'll highlight the one with 10000 examples. In general, the original dataset has about an even spread of highly rated movies and lowly rated movies (Mass et al., 2011). This observation remains seemingly true in our dataset as Figure 1 highlights how the level of disparity between the occurrences of each rating is small. Given how we defined a 'positive' and 'negative' sentiment on the basis of the ratings, Figure 1 presents how there is close to an even amount of positive and negative reviews with the number of positive reviews being more as detailed in Figure 3. This observation is further reflected in Figure 2 where each emotion has a bit more positive reviews than negative ones with the exception being 'surprise'. Overall, the figures showcase how the data is not heavily skewed towards one binary sentiment which serves well when running SVM. On the other hand, it is apparent that 'sadness' dominates heavily in the spread of emotions which may cause concerns when running RNN. Another thing to note is that the vocab size after vectorizing the data is 7508.



```
Number of Reviews:  10000
Average Word Size in a Review:  929.5481
Distribution of Labels:  Counter({1: 5802, -1: 4198})
Vocabulary size from text vectorization:  7508
```

Figure 3: Data statistics for the subset of size 10000.

The RNN model classifies the reviews into emotions, based on a lexicon that maps the words in the movie descriptions to the corresponding emotion label. Initially, the model preprocesses the descriptions by tokenizing them and padding sequences to a maximum length. The embedding layer converts words into dense vectors. The subsequent GRU layers with 128 and 64 units, respectively, process the sequences, while dropout layers with a 20% dropout rate help prevent overfitting. Two dense layers, with 32 units each and a softmax activation function in the final layer, enable multi-class emotion classification. Training employs the Adam optimizer and categorical cross entropy loss, with accuracy as the evaluation metric. The model is trained on the descriptions and corresponding emotion labels encoded as one-hot vectors. A portion (40%) of the training data serves for validation, ensuring the model's generalization ability. Finally, the model's performance is assessed on the movie reviews. An RNN is particularly useful for this task due to its ability to capture sequential dependencies inherent in language data, which will allow us to compare the emotion expressed in the reviews versus the actual intended emotions.

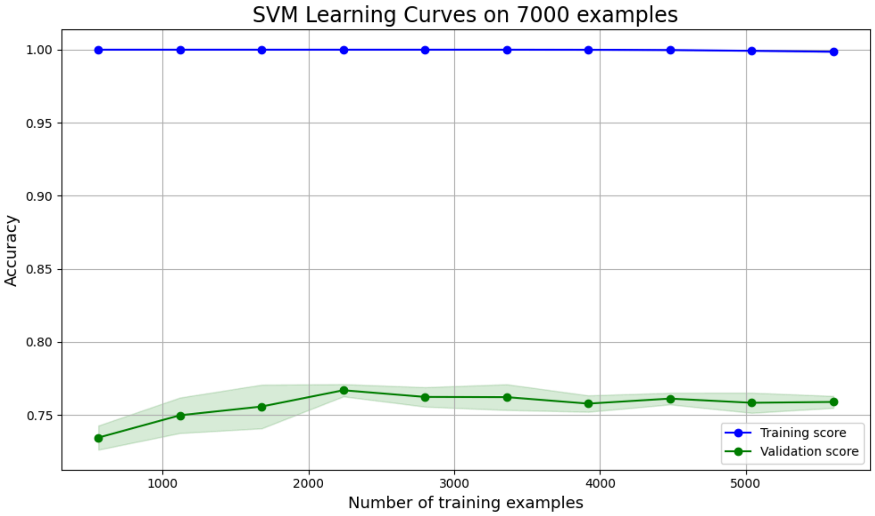


Figure 4: SVM Training Curves

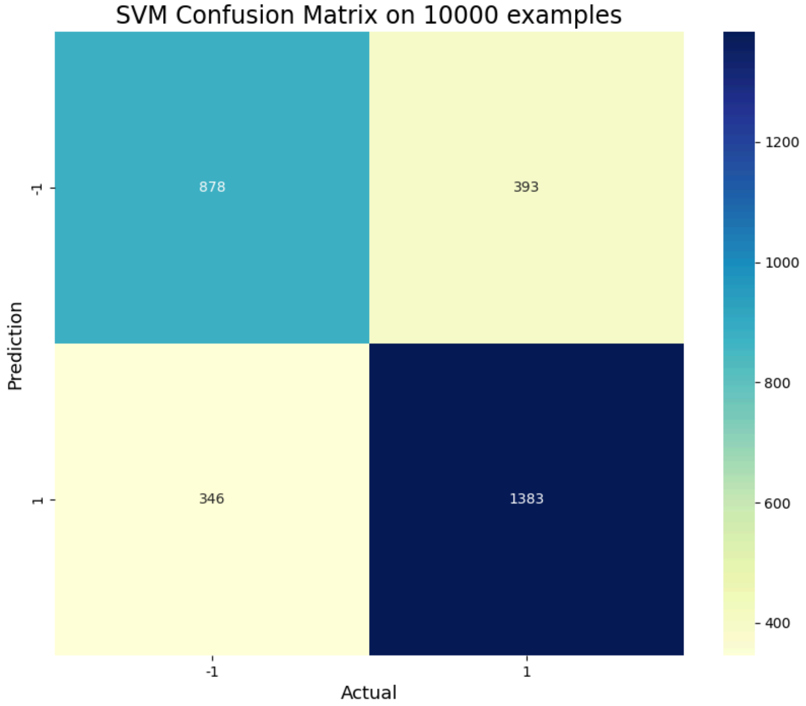


Figure 5: SVM Confusion Matrix: 1=pos, -1=neg sentiment

When running SVM, we first measured how well the model will be able to generalize the data using 5-fold cross validation. Figure 4 presents the training curves for SVM when using 70% of the 10000 data subset as train data. Average score from performing cross-validation is 0.7589. Overall, considering that there are two classes for our SVM model being a label of 1 or -1 for pos or neg sentiment, SVM has shown to be able to classify the data well with the average accuracy score being above 50%. This observation is then justified by Figure 5 which highlights the confusion matrix after fitting SVM on the train data and using it to predict the test data. Both the true positive and true negative values are relatively high with the false values being more on the lower end. Because of vectorization, each word present in the vocabulary has an attached weight in the SVM model. Using this data, we then plotted the top 10 words that denoted a positive or negative sentiment.

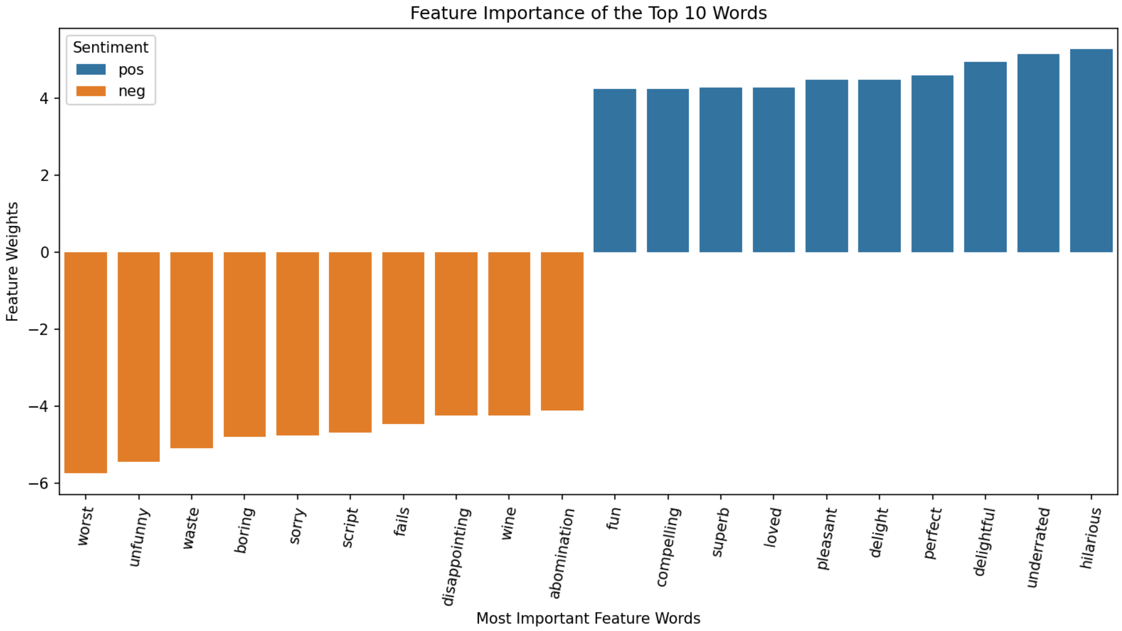


Figure 6: Top 10 most important words in the 10000 data subset

The training and validation datasets were subsets of the movie descriptions, while the test data comprised reviews. In Figure 7, the RNN exhibited impressive performance, achieving 98% training accuracy after 5 epochs and 97% validation accuracy after 3 epochs. However, beyond 10 epochs, the model showed signs of overfitting on the training data, with negligible improvement in validation accuracy. When trained on the entire dataset, the RNN attained a test accuracy of approximately 21%, evaluated on 19316 unique reviews. Given the threshold of 1/8 for 8 emotion classes, this performance is notably robust. Figure 8 illustrates the confusion matrix, revealing that the model accurately predicted anger most often, while surprise was the least accurately predicted emotion. This disparity underscores the nuanced nature of emotion perception in movie reviews, where only 21% align with the intended emotions depicted in the movie descriptions.

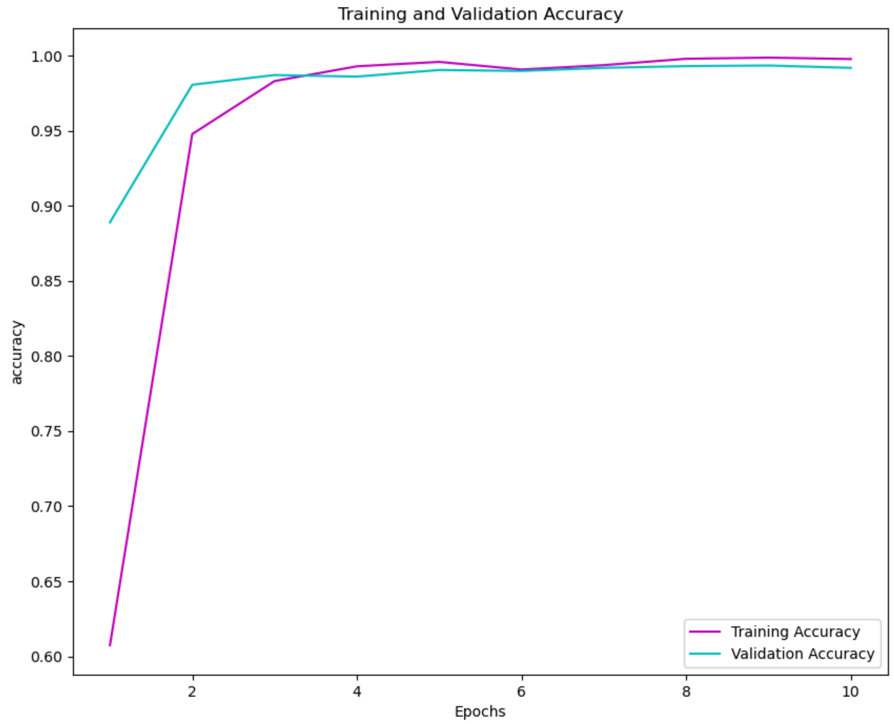
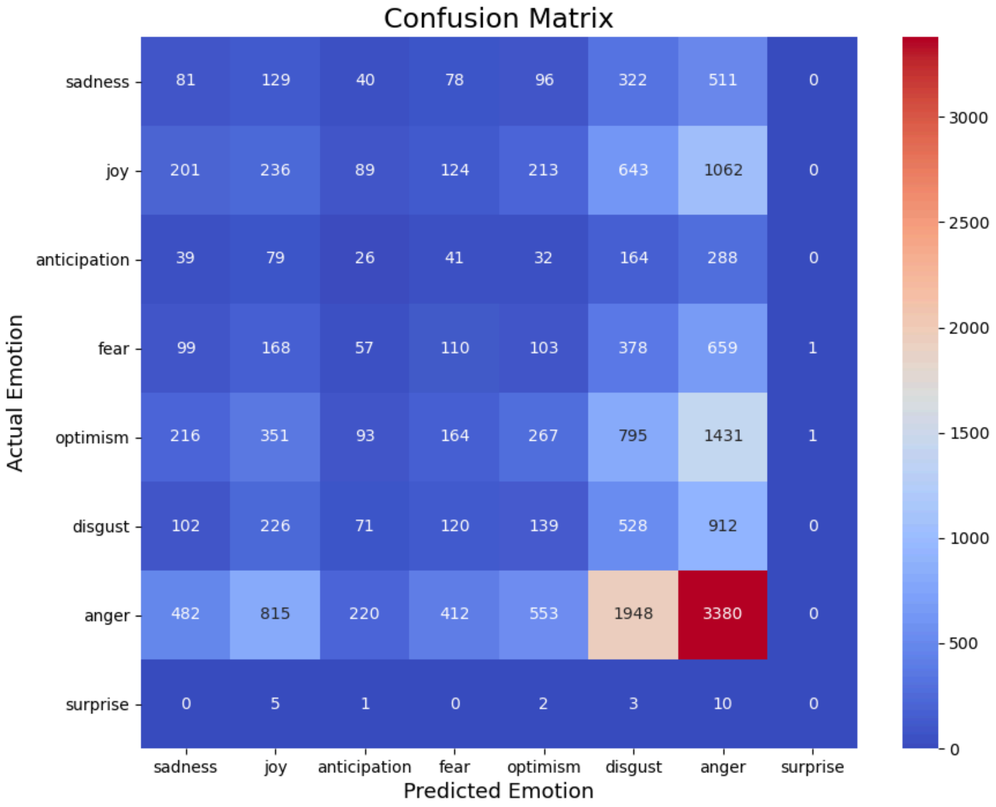


Figure 7: Learning curve for training and validation accuracy



	sadness	joy	anticipation	fear	optimism	disgust	anger	surprise
sadness	81	129	40	78	96	322	511	0
joy	201	236	89	124	213	643	1062	0
anticipation	39	79	26	41	32	164	288	0
fear	99	168	57	110	103	378	659	1
optimism	216	351	93	164	267	795	1431	1
disgust	102	226	71	120	139	528	912	0
anger	482	815	220	412	553	1948	3380	0
surprise	0	5	1	0	2	3	10	0

Figure 8: confusion matrix for RNN

In conclusion, our project on performing sentiment analysis on movie reviews sought to classify reviews into binary sentiments such as 'positive' or 'negative', and emotions like 'joy' or 'sadness'. Recognizing the potential disparity between the sentiments portrayed by reviewers and the intended emotions depicted in movie descriptions, we developed two models: a Support Vector Machine (SVM) and a Recurrent Neural Network (RNN). Our data, sourced from the IMDb Movie Reviews dataset, underwent extensive preprocessing to prepare it for analysis. The SVM focused on binary sentiment classification, while the RNN aimed to classify emotions based on movie descriptions. Despite the challenges posed by the dominance of 'sadness' in the emotion spread, the RNN demonstrated impressive performance, achieving 98% training accuracy after 5 epochs. However, the test accuracy of approximately 21% highlighted the nuanced nature of emotion perception in movie reviews. In contrast, the SVM showcased robust classification capabilities, with an average accuracy score above 50%. For future work, we aim to explore more advanced methods to enhance the accuracy and robustness of sentiment analysis on movie reviews. One framework is to develop a single model that integrates a binary classifier to distinguish between positive and negative sentiments, which can then further classify the reviews into one of the eight emotion classes. Additionally, we plan to leverage ensemble methods to identify the most effective classifiers for binary sentiment classification and explore the classification of emotions into the eight classes. Another direction is to investigate the use of transformers to generate reviews based on movie names, genres, and descriptions, thereby enhancing the understanding of audience perceptions and preferences. Furthermore, we intend to utilize pre-made lexicons that map emotions to words to conduct more authentic sentiment analysis, providing deeper insights into the emotional nuances conveyed in movie reviews.

References:

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. (2011). Learning Word Vectors for Sentiment Analysis. *The 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011).*

Data: IMDb_Movie_Reviews_Genres_Description_and_Emotions (kaggle.com)

Santosh Kumar Bharti, S Varadhaganapathy, Rajeev Kumar Gupta, Prashant Kumar Shukla, Mohamed Bouye, Simon Karanja Hingaa, Amena Mahmoud, "Text-Based Emotion Recognition Using Deep Learning Approach", *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 2645381, 8 pages, 2022. https://doi.org/10.1155/2022/2645381