Analyzing Sentiment of Online Film Reviews

Michael Rosanelli, Chris Veilleux, Jackson Douma

Computer Science Department, Lakehead University Orillia

mlrosane@lakeheadu.ca

cjveille@lakeheadu.ca

jcdouma@lakeheadu.ca

***Abstract*— Performing sentiment analysis on movie reviews deepens our understanding of how to quantify the emotional tone of written human language. This report aims to evaluate the accuracy of 4 machine learning classifiers and their various input features in predicting the sentiment of these reviews. After preprocessing a dataset of movie reviews and their associated sentiment ratings [1], relevant features were engineered to assess aspects of each review that could contribute to its rating. Using forward feature selection and a TF-IDF matrix, 4 machine learning classifiers were trained: K-Nearest Neighbours, Random Forest, Decision Tree, and Support Vector Machine. All model accuracies were computed, and various visualizations were created to evaluate the importance of selected features and overall model performance. All models returned moderately high accuracy scores ranging from 72.05% (Decision Tree) to 88.07% (Support Vector Machine). These models demonstrate the effectiveness of machine learning in sentiment analysis, and this report exposes various areas for improvement to increase model accuracies.**

***Keywords*— Sentiment analysis, lemmatized, VADER, forward feature selection**

I. Introduction

Sentiment analysis refers to a multi-stage process employing natural language processing techniques to identify the emotional tone of a body of text as positive or negative. Sentiment analysis is highly relevant in business today as it can help businesses understand customer opinions and then use that data to improve the experience they offer. Beyond the world of business, sentiment analysis has many applications including understanding public perception of political candidates, analyzing user feedback to develop new product features, enabling automated systems like chatbots to make data-driven decisions, and so on.

In this paper, we explore each stage of the sentiment analysis process to develop the most accurate machine-learning model possible. Section II outlines the data preprocessing work completed, transforming the disorganized source data into a well-formatted .csv file ready for feature extraction. This section also describes our feature engineering process, providing details for each feature and discussing how forward feature selection and a TF-IDF matrix were used to determine the best-performing features. Finally, an overview of the machine learning models trained is provided. Section III details the results, touching on the accuracy of each model and describing the effect of review length, quantity of exclamation marks, and the effect of various words and phrases on model performance, and a further discussion of model accuracies. Multiple data visualizations are also provided and discussed. Section IV concludes our research findings and reiterates the importance of this topic.

II. Methods

A. Data Janitor Work

Due to the atypical format in which the source data [1] was provided, a large amount of janitor work was required. Each review was its own text file, located in subfolders ‘pos’ or ‘neg’ of the ‘train’ folder, with a total of 24,186 reviews to use in our finalized dataset.

To create a usable dataset for our project, the first step was to analyze the path of each text file and add a 0 or 1 to a new .csv file, depending on whether the file was located in the ‘neg’ or ‘pos’ subfolder. Then, the contents of the text file containing the text of the film review were appended. Lastly, the star rating of the review (a score from 1 to 10) was extracted from the file name and appended to the .csv file, leaving a 3-column file.

Next came the cleanup step, which involved creating a new .csv file with all stop words, line breaks, and lone punctuation removed from film review text, as well as lemmatizing each word and converting it to lowercase. This new .csv contains the class label (0 or 1), the preprocessed review text, the review's star rating, and the number of exclamation points in each review, and it is the file used for the remainder of our project development.

B. Feature Engineering

The features used for this AI model were used to detect the sentiment of a given set of words, in this case, a film review. There are ten original features, four of which were generated through forward feature selection:

1. Amount of positive words: Check how many words in the review have positive sentiments. This feature also uses the stemmed version of each word in the review. By seeing how many words of the review are in an array of lemmatized positive words called positive_words, a score => 0 will be generated to give a positive score.
2. Amount of negative words: Check how many words in the review have negative sentiments. This feature also uses the stemmed version of each word in the review and uses the same process as the positive score, but with an array of lemmatized negative words called negative_words, a score => 0 will be generated to give a negative score. The score will also increment if a ** is detected, to check for harsh profanity use.
3. Amount of reverse sentiment: checking for any time an "only" appeared in the review, and incrementing a score based on any following words. Once again, this feature also uses the stemmed version of each word in the review. Since people often say things like "the only bad thing" or "the only decent aspect" in reviews, the previous two features flag the sentiment as the opposite of what we want. In this case, a score would be incremented if a string from positive_words was found in the following three words after "only", or decremented if a string from negative_words was found in the following three words. Unless the string preceding "only" is "not" (to account for phrases that start with "not only is…"), then it will have the opposite effect on the score.
4. Adverb-to-adjective ratio: After seeing a paper [2] and a blog post by Professor Mark Liberman from the University of Pennsylvania [3] which talk about using adverbs and adjectives in tandem when doing sentiment analysis, we decided to take inspiration from that and compare the number of adverbs to the number of adjectives in each review, which each word has again been stemmed. By looping through the review and using part-of-speech tagging from the nltk Python library, we counted each number and returned a ratio in the form of a decimal number.
5. Average VADER score of all nouns: By utilizing the VADER sentiment analysis tool, which is also included in the nltk Python library and part of speech tagging, we can generate the average score that VADER gives to different types of words. The "compound" score returned by the "polarity_scores" method will result in either a positive or negative value that corresponds to the word passed in—this feature in particular checks for the average of all the scores of each noun in the review.
6. Average VADER score of all adjectives: This is the same process as feature 5 but regarding each adjective in the review.

ML Classifier	Forward Selected Features			
	Feature 1	**Feature 2**	**Feature 3**	**Feature 4**
K-Nearest Neighbours (k = 3)	Positive Word Count	Negative Word Count	Noun Vader Score	Adjective Vader Score
Random Forest	Positive Word Count	Negative Word Count	Reverse Sentiment	Exclamation Point Count
Decision Tree	Positive Word Count	Negative Word Count	Reverse Sentiment	Exclamation Point Count
Support Vector Machine	Positive Word Count	Negative Word Count	Noun Vader Score	Adjective Vader Score

TABLE I
FORWARD SELECTED FEATURES

7. Average VADER score of all verbs: Same process as features 5 and 6, but with the verbs found in the review.
8. Average VADER score of all adverbs: Once again, it is the same process as the previous three features, this time examining the adverbs.
9. Length of the review: By using the len method, we can include the length of a given review.
10. Amount of exclamation points: During the pre-processing phase before removing any of the punctuation, the number of exclamations was examined for each review. Using this feature to measure the intensity of sentiment was the idea.

These ten features went through forward feature analysis using the mlxtend Python library and were tested against each of the four classifiers that were used. After performing the analysis, the curated features were combined with a matrix of TF-IDF scores for all the reviews to create a two-dimensional array containing all features that would be used to train the machine learning classifiers listed in Table 1. The features for each classifier are the ones that were determined to return the best results by the feature-selection algorithm from the mlxtend library.

III. RESULTS

This section presents the outcomes of the analysis done with feature engineering. It examines the results of the different models and presents the most interesting data that was found relating to common aspects that make a review positive or negative. This includes length of review, exclamation marks used, words, and phrases.

A. Model Accuracy

The 4 types of models used. K-Nearest Neighbours got to 75.28%, Random Forest got to 84.76%, Decision Tree got to 72.05%, and Support Vector Machine got to 88.07%. [Table II]

There was a noticeable improvement in some model accuracies after feature selection was added. K-Nearest Neighbours saw an improvement of 4.8%, and Support Vector Machine improved by 3.45%. Random Forest and Decision trees also improved, but the amount was rather minimal, with both being below 0.5%.

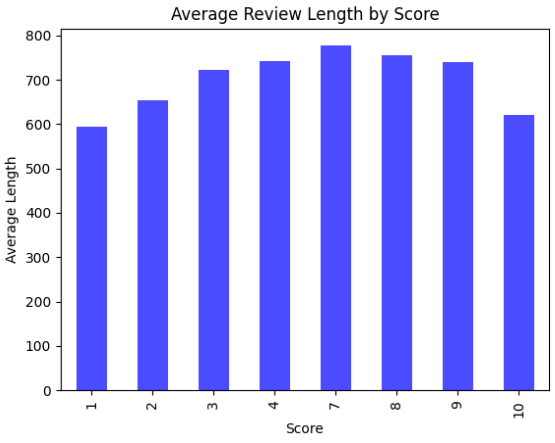

Fig 1. A bar graph of average review lengths sorted by score.

B. *Review Length*

One method we used is attempting to find the correlation between the length of a review and its positive or negative. [Table III]

Overall, the negative and positive reviews seem to follow a similar pattern. However, there is a clear spike in negative reviews compared to positive reviews in the 400-600 characters area, while positive reviews have slightly more reviews almost everywhere else. These results seem to suggest that people who are passionate about a movie seem to write more on average, whereas people leaving a negative review tend to keep it brief.

We can look at this even further by calculating the exact review length for each score from 1-10. Looking at Figure 1, you can see a steady increase in review lengths from 1-9, with a large decline in 10 score reviews. The decrease in scores of ten is likely due to those people simply leaving a perfect score with no elaboration, whereas people leaving a specific number are much more likely to analyze and elaborate.

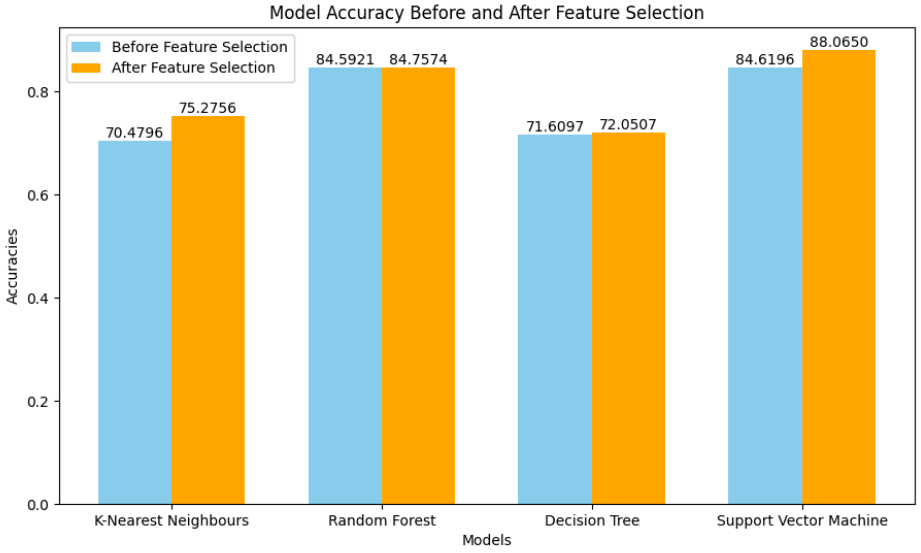

TABLE II

MODEL ACCURACY BEFORE AND AFTER FEATURE SELECTION

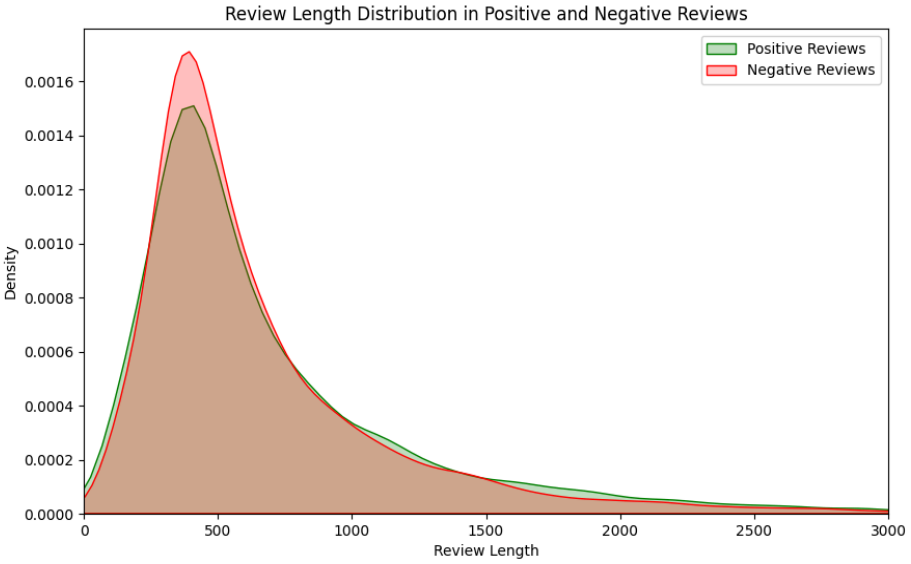

TABLE III

REVIEW LENGTH DISTRIBUTION OF POSITIVE AND NEGATIVE REVIEWS

C. *Exclamation Marks*

Another idea is to examine the amount of Exclamation Marks in positive and negative reviews, to measure the intensity of a sentiment.

There is a clear downward trajectory that indicates a decrease in score if there are exclamation marks [Figure 2].

This is further backed up by Table IV, which shows that reviews with no exclamation marks are more likely to be positive, and reviews with one or more exclamation marks are more likely to be negative.

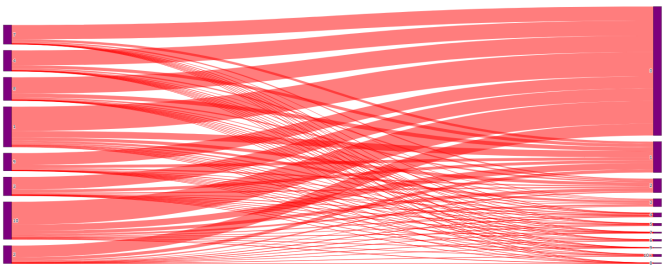

Fig 2. A Sankey Diagram comparing the number of exclamation marks on the left, and the score on the right.

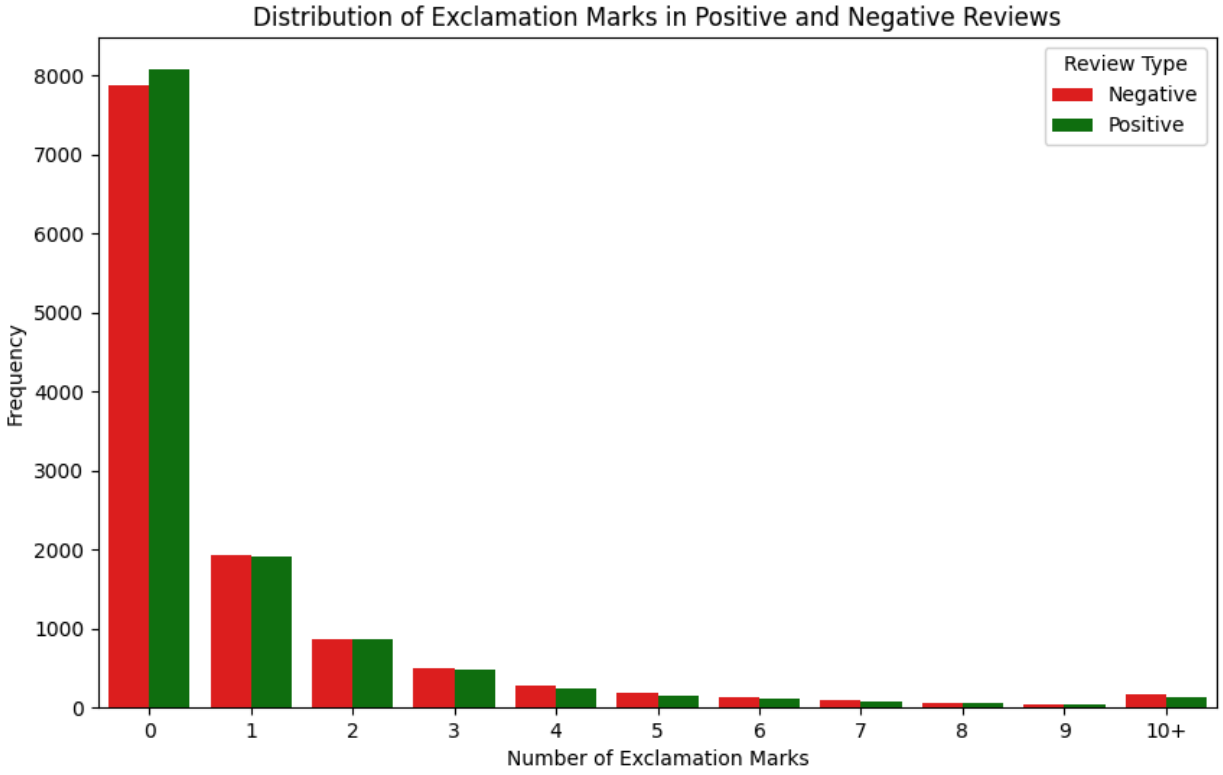

TABLE IV

DISTRIBUTION OF EXCLAMATION MARKS IN POSITIVE AND NEGATIVE REVIEWS

D. Words and Phrases

Using the actual words and phrases of the reviews is another thing we could use. For the following data, the more uninteresting words that don't mean anything for our purposes like "movie", "film", "thing", or "come" have been removed. [Table V]

Table five shows both very predictable, and some more unique results. To be expected, words like "good" are high on the positive list, and words like "bad" are high on the negative list. "Character" and "story" are high on both lists, which makes sense as they are crucial to making a good movie. However, a more interesting word on the list "actor" is much higher on the negative list than the positive list. This suggests that a poor actor can affect the score of a movie much more than a good actor.

Phrases is another interesting statistic that was tracked, that yielded some more meaningful results. For the following, the amount of consecutive two words together was tracked, and the top twenty were displayed. [Table VI].

This data uncovers a lot of things that we didn't have access to before. Phrases like "special effect" and "main character" are much higher in negative than positive reviews, which suggests that these things can hurt a movie a lot if they are poor, but not necessarily make it an amazing movie if they are good. A very unexpected positive high phrase that is not in negative reviews is "new york", this could be due to good movies being made or taking place in it being enjoyed by people much more than other cities. Although our dataset didn't have access to the genre, phrases were able to uncover that "horror film" is in the negative review top twenty, and "love movie" and "fall love" are in the positive review top twenty, which shows that people are much more likely to enjoy a romance movie than a horror movie.

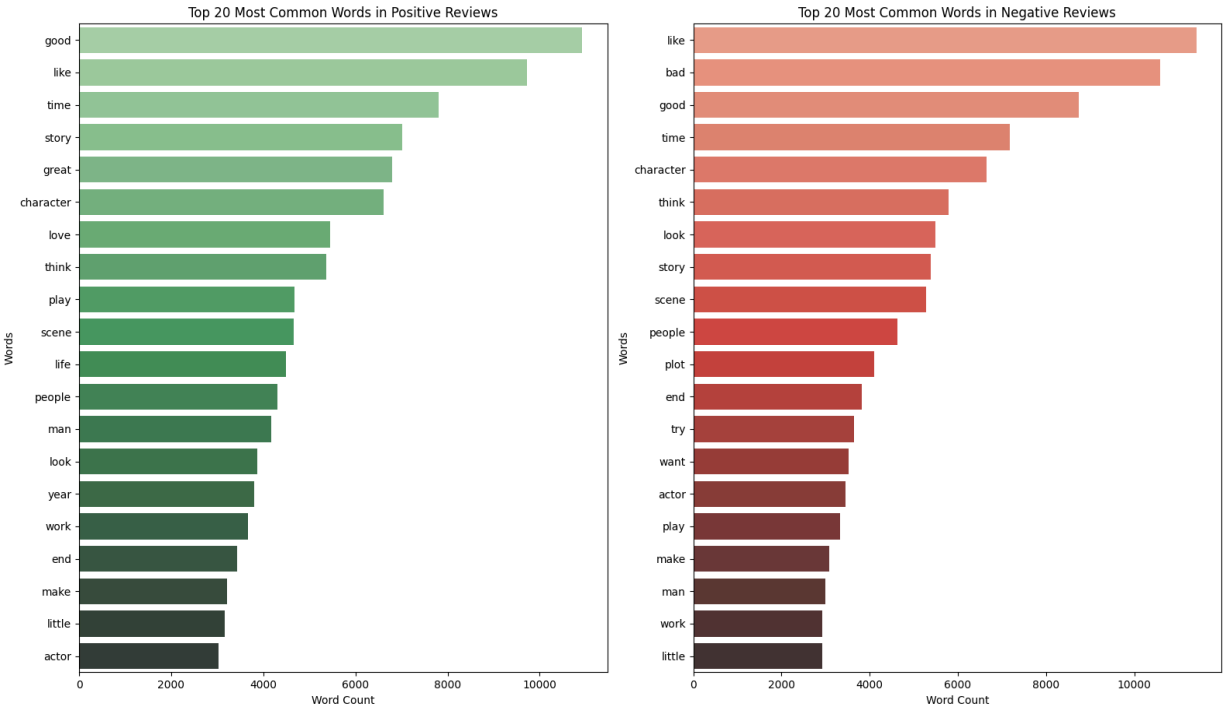

TABLE V

TOP 20 MOST COMMON WORDS IN POSITIVE AND NEGATIVE REVIEWS

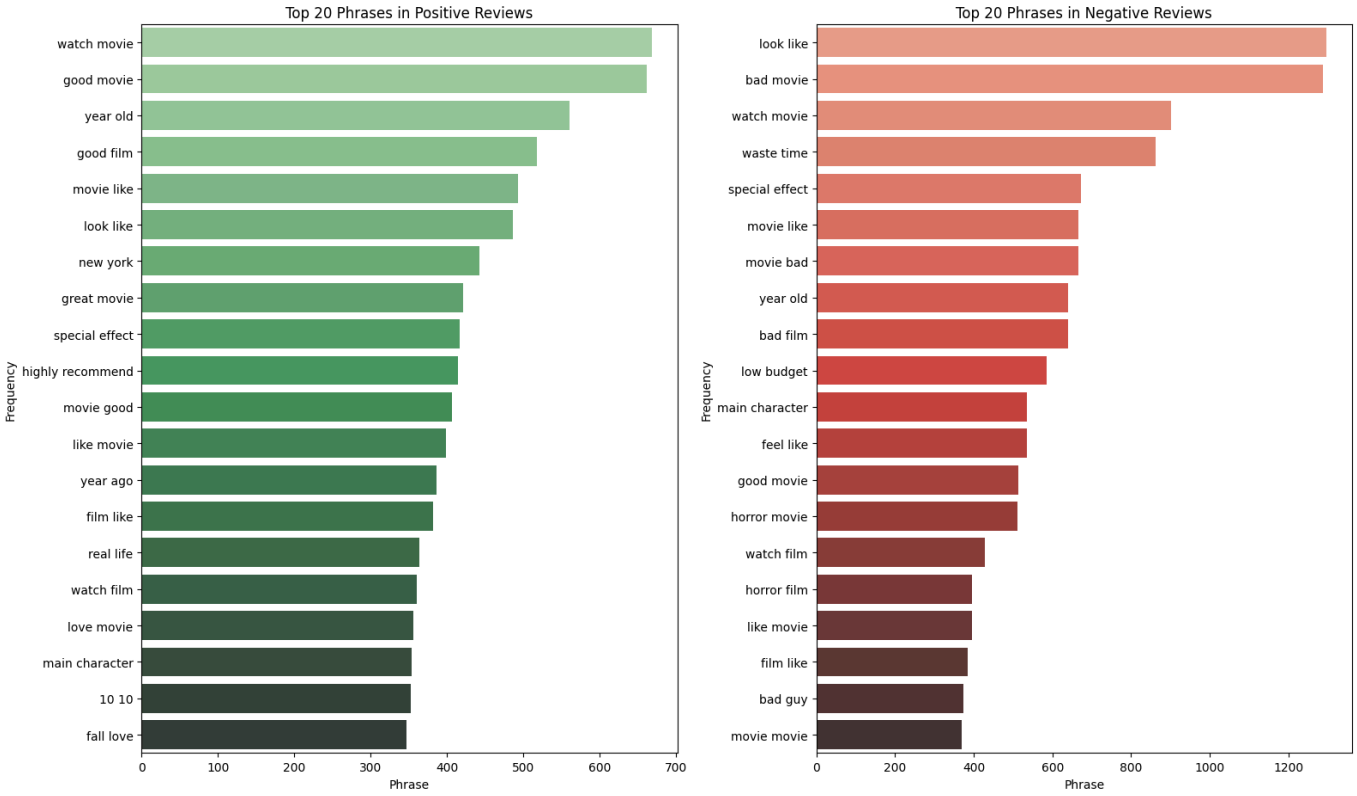

TABLE VI

TOP 20 MOST COMMON PHRASES IN POSITIVE AND NEGATIVE REVIEWS

E. Final Results

The final conclusion to this project results in the Support Vector Machine classifier being the most reliable at accurately predicting positive or sentiment from a film review, with the Random Forest at a close second place. Figures 3 through 6 are confusion matrices regarding the performance of each of the four supervised learning classifiers. Regarding our choice of classifier, the SVM returned a precision value of 0.893, and a recall of 0.879. It's interesting to note that each classifier except for the Random Forest had more false positives than false negatives, and actually had the largest disparity in the SVM. To conclude, the final result of this project is the Support Vector Machine classifier that was trained to detect sentiment.

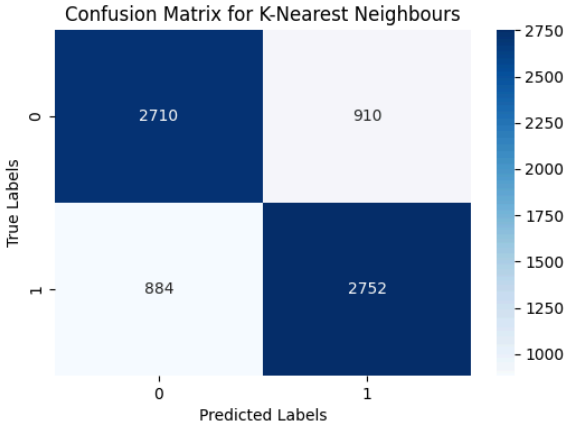

Fig 3. A confusion matrix for the K-Nearest Neighbours model

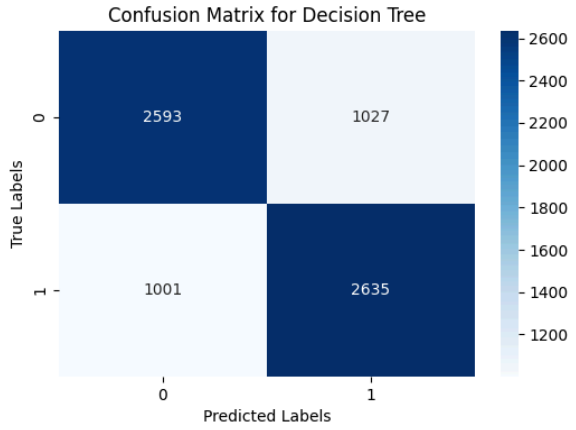

Fig 4. A confusion matrix for the Decision Tree model

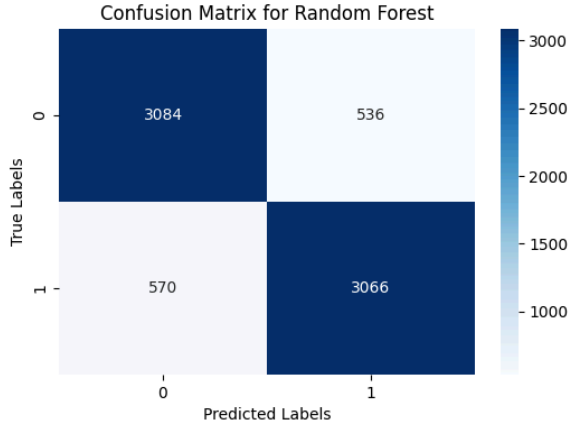

Fig 5. A confusion matrix for the Random Forest model

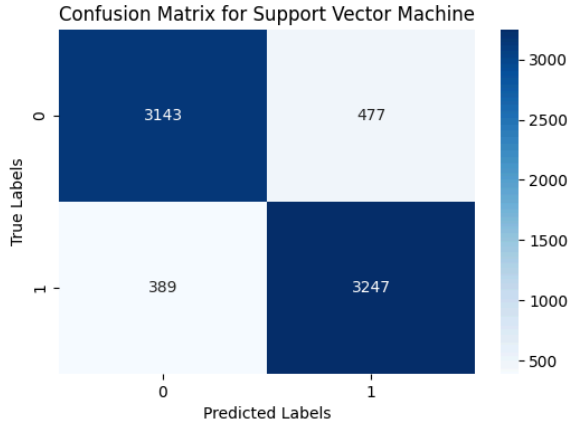

Fig 6. A confusion matrix for the Support Vector Machine model

IV. Conclusions

By engineering relevant features and evaluating them, training a variety of models, and exploring the results through metrics, visualization, and discussion, our project provides an overview of machine learning techniques, natural language processing, and their effectiveness in analyzing the sentiment of movie reviews. We achieved the greatest accuracy, 88.07%, using a Support Vector Machine model with 4 features chosen through forward selection: positive word count, negative word count, noun Vader score, and adjective Vader score. Despite achieving a moderately high accuracy score, we recognize the need for continual research in this area, as human language contains complexities like sarcasm, irony, slang, or context-dependent meaning which our features do not directly address. Incorporating features that account for such nuances would serve to not only improve model accuracies but also support the ultimate goal of accurate measurement of emotional tones in written human language.

Acknowledgment

We thank Dr. Chris Brogly for teaching us about artificial intelligence models, using classifiers to predict data, and imparting all other kinds of knowledge for us to use during this project.

We also wish to acknowledge Michael Shell and other contributors for developing and maintaining the IEEE LaTeX style files which were used for this report.

References

[1] Andrew Lukyanenko. “Movie Review Sentiment Analysis EDA and models.” Kaggle. https://www.kaggle.com/code/artgor/movie-review-sentiment-analysis-eda-and-models/notebook (accessed October 29, 2024).

[2] F. Benamara, C. Cesarano, A. Picariello, D. R. Recupero, and V. S. Subrahmanian, “Sentiment Analysis: Adjectives and Adverbs are better than Adjectives Alone,” *International Conference on Weblogs and Social Media*, pp. 203–206, Jan. 2007.

[3] M. Liberman, “Language Log» Adjectives and adverbs,” *Upenn.edu*, 2016. https://languagelog.ldc.upenn.edu/nll/?p=25782 (accessed Nov. 25, 2024