CIS 9660 - Applied Natural Language Processing

Instructor: Chaoqun Deng
Zicklin School of Business, Baruch College, CUNY

Text Analysis of The Movie Reviews Corpus

Team 2: Anisa Rashid, Alessandro Sciorilli, Myar Zaid, Yanjambou (Yamou) Jagne

May 14, 2024

Abstract

In an effort to analyze a dataset of movie reviews, we utilized a number of text analytical approaches as we defined eight different functions for feature extraction, split the dataset into training and test sets, and observed accuracy, precision, recall, and F1 scores of each machine learning model to assess the best performer in terms of classification/prediction.

1 Dataset Description

The dataset we used for this analysis is the Movie Reviews Corpora from the NLTK Corpus. The Movie Reviews Corpora can be accessed from the NLTK corpus and it contains a collection of 2,000 movie reviews which is sourced from the internet movie database. Each movie review is stored as a text file and is labeled based on their sentiment, thus either negative or positive. Upon running the len() function, we were able to conclude that there are equal amounts of negative and positive reviews in the Movie Reviews Corpora so 1,000 positive reviews and 1,000 negative reviews.

2 Research question

Our research question aims to discover the textual features that contribute to the classification of movie reviews into positive or negative sentiment categories. This research question is important as it will allow us to understand how the choice of words people tend to use when writing positive versus negative reviews differs. In having this understanding we are able to grasp how emotions and sentiments are conveyed through language.

3 Text Preprocessing

In order to ensure the utmost accuracy and quality in our data, we conducted preprocessing of our data. We converted all text to lowercase to ensure consistency and reduce vocabulary size. We also eliminated non-alphanumeric characters, punctuation marks, numbers, and symbols, as they often don't contribute much to the analysis. Adding to this we removed common words like "the", "is", "and", etc., which are frequently occurring but typically do not carry much semantic meaning. Lastly, we did tokenization as this would split the text into individual words/tokens for further analysis.

4 Text Analytic Method Description and Results

The features which we extracted entailed extracting the 2000 Most Frequent Words, 2000 most frequent bigrams, 2000 Most Frequent Trigrams, 1000 Most Frequent Words in Positive Reviews that are not in Negative Reviews, and vice versa, Unique Words Contained in the Positive/Negative Lexicon Database, Frequency of Emotional Language In Positive and Negative Review, Topic Models, and TF-IDF. Once we defined the features to extract, we made sure to split the dataset into training and test sets in order to begin evaluating the various machine learning models for these features. We used 70% of the data as a training set and the remaining 30% as a test set. The machine learning models we used included Naïve Bayes (both Gaussian and Bernoulli), Logistic Regression, Nearest Neighbors, Random Forest, Decision Tree, Support Vector Machine (with kernels: Linear, Polynomial, Radial Basis Function (RBF), and Sigmoid), Neural Net, and AdaBoost. In order to access all models with ease we created a list of the models and their functions called `model_list`. Overall, we evaluated the performance of each machine learning model with a classification report to assess accuracy, precision, recall, and F1-score.

4.1 Feature 1: 2000 Most Frequent Words

4.1.1 Why We Chose This Text Analytic Approach

For our first feature, we chose to analyze the 2000 most frequent words in the text corpus. This approach allowed us to get an idea of what our most frequent (2000) words are, which helps with text

classification and getting a more detailed context of our dataset and specifically the words which are used in positive versus negative reviews.

4.1.2 How To Use This Method

In order to use this method, we implemented a function that calculates the frequency of each word in a given document. This function iterates through each word in the document and updates a dictionary with the count of each word. We then applied this function to each text in the dataset, resulting in a series of dictionaries where keys represent words and values represent their frequencies. Then, we converted this series of dictionaries into a dataframe, where each column corresponds to a word and each row corresponds to a document in the dataset. Finally, we included a 'sentiment' column to indicate the sentiment label associated with each document.

4.1.3 Results

All in all, our result from this approach was a dataframe which is like a snapshot of all the movie reviews, where each row captures a different review, and every column represents a specific word that appears in those reviews. The numbers in the dataframe show how often each word pops up in each review, giving us a glimpse into the words that keep popping up across different pieces of feedback. Not only this, but given that there is a sentiment column at the end which tells us whether the review is positive or negative we can assess how the frequency of certain words contribute to the sentiment of a movie review.

4.1.4 Model Evaluation

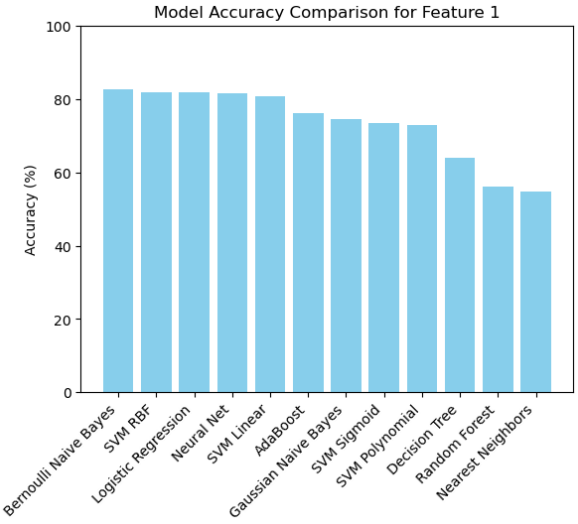

Figure 1: Model accuracy comparison for feature 1

Upon evaluating each model for this feature, we found that the top-performing model was the Bernoulli Naive Bayes, performing well in accuracy, recall, and F1 score. It achieved an accuracy of 82.50%, a recall of 77.93%, with a precision rate of 84.64%. Conversely, the Nearest Neighbors model (KNN) performed poorly, displaying only a 54.83% accuracy and a very low recall of 18.28%, alongside a precision of 60.92%.

4.2 Feature 2 - 2000 Most Frequent Bigrams

4.2.1 Why We Chose This Text Analytic Approach

For our second feature we chose to analyze the 2000 most frequent bigrams because it provides a deep understanding of how words are commonly paired together. With this understanding, we can identify

recurring phrases, expressions, and linguistic features specific to movie reviews. This also can aid in sentiment analysis by identifying bigrams associated with positive or negative sentiment.

4.2.2 How To Use This Method

In our effort to use this method we began by defining a function to generate bigrams from a list of words and then iterating through each text in the dataset, counting the frequency of each bigram. The top 2000 most frequent bigrams are selected and used to create a feature for each text. This captures the frequency of occurrence of each top bigram in the text. Finally, the script converts all this into a data frame, including the sentiment labels given to each text. By doing so, we can gain insights into the language patterns and sentiments expressed in movie reviews.

4.2.3 Results

All in all, our result from this approach was a dataframe containing all the reviews where each column in the dataframe represents individual bigrams extracted from the movie reviews. The rows represent individual movie reviews, where each cell in the data frame contains a binary value indicating whether the corresponding bigram appears in the respective review. Thus, a '1' in the column meant that the corresponding bigram was found in the document, and '0' meant it was not. Moreover, the last column contains sentiment labels related to the movie reviews to give us a better understanding of whether it's positive or negative.

4.2.4 Model Evaluation

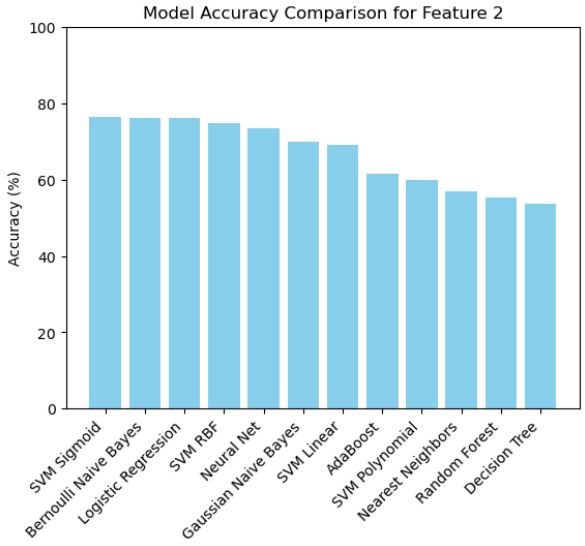

Figure 2: Model accuracy comparison for feature 2

Upon evaluating each model for this feature, we found that the top-performing model was the SVM Sigmoid, performing well in accuracy, recall, and F1 score. It achieved an accuracy of 76.33% and a recall of 79.31%, with a precision rate being lower at 73.72%. Conversely, Decision Tree model is the worst performer, displaying only a 53.67% accuracy and a precision of 51.07%.

4.3 Feature 3: 2000 Most Frequent Trigrams

4.3.1 Why We Chose This Text Analytic Approach

For our third feature, we chose to analyze the 2000 most frequent trigrams which refer to sets of three consecutive words that occur most frequently in a text. Analyzing these frequent trigrams can offer a deeper understanding of the structures of the language. By identifying recurring trigrams, we can

enhance our ability to interpret the context behind the movie reviews and better understand why a movie review is categorized positive or negative.

4.3.2 How To Use This Method

In our effort to use this method, we began by defining a function to generate trigrams from a list of words and then iterating through each document in the dataset, counting the frequency of each trigram. The top 2000 most frequent trigrams are selected and used to create a feature for each document. This feature captures the frequency of occurrence of each top trigram in the document. Then, the script applies this feature extraction function to each text in the dataset, creating a new column 'features' containing the trigram features. Finally, it converts this feature representation into a data frame, including the sentiment labels associated with each document.

4.3.3 Results

All in all, our result from this approach was a dataframe with all the movie reviews where each row corresponds to a specific review and each column represents a trigram. The values in the data frame indicate whether a particular trigram appears in each review, with 1 indicating its presence and 0 indicating its absence. In addition to this, we have a sentiment column that contains labels indicating the sentiment of each review.

4.3.4 Model Evaluation

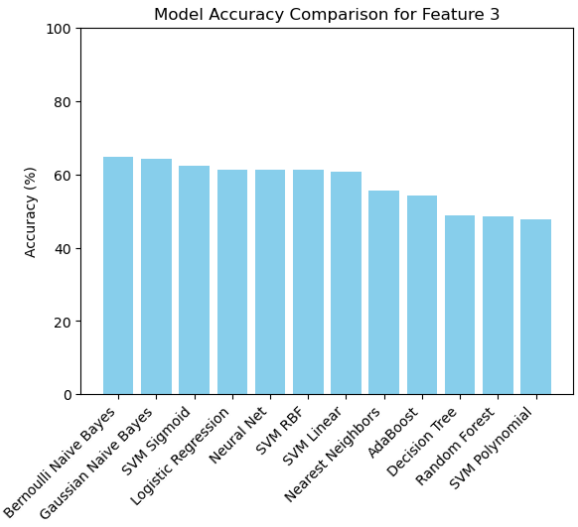

Figure 3: Model accuracy comparison for feature 3

Upon evaluating each model for this feature, we found that the top-performing model was the Bernoulli Naive Bayes. It performs highest in accuracy (64.83%) and Precision (65.02%), but more poorly on recall (58.97%), resulting in an F1 score of 61.84%. On the other hand, the SVM Polynomial was the worst performing model, displaying an accuracy of 47.67% with a precision of (47.86%) but very high recall (92.76%), leading to a F1 score of 63.15%.

4.4 Feature 4: 1000 Most Frequent Words in Positive Reviews That Are Not in Negative Reviews, and Vice Versa

4.4.1 Why We Chose This Text Analytic Approach

For our fourth feature, we chose to analyze the 1000 Most Frequent Words in Positive Reviews that are not in Negative Reviews, and Vice Versa. This approach was very important to us in our analysis as it allowed for us to understand the difference in words which people tend to use in positive reviews

versus negative reviews. This understanding is different from just extracting the most frequent words in positive and negative reviews because when you do that there may be the case where some of the words used in the positive and negative reviews are the same.

4.4.2 How To Use This Method

In order to use this method we began by creating two lists to hold words from positive and negative reviews. We then split the words into positive and negative lists based on the sentiment with which the reviews were labeled. Following this step we counted the frequencies of words in both the `pos_words` or `neg_words lists` . Once we had the frequency count, we removed the words that both dictionaries had in common. We then selected the top 1000 words from both positive and negative that are unique to each sentiment and then we defined a function to create binary features for each document. Our final step in executing this approach was to apply the function to each text in the dataframe and convert the series of dictionaries to a dataframe which included the sentiment column.

4.4.3 Results

All in all, our result from this approach was a dataframe containing all the reviews where each column in the dataframe was a unique word which the `pos_words` and `neg_words` did not have in common. A '1' in the column meant that the corresponding word was found in the review, and '0' meant it was not. That said, with the sentiment of each review being the last column, we were able to understand the distinctive language patterns that differentiate positive reviews from negative ones.

4.4.4 Model Evaluation

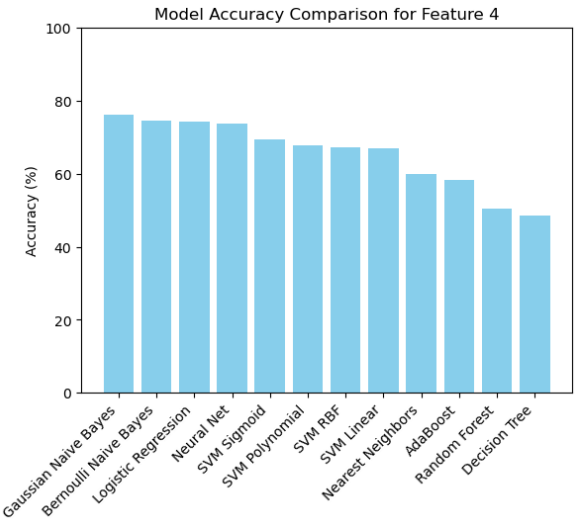

Figure 4: Model accuracy comparison for feature 4

Upon evaluating each model for this feature, we found that the top-performing model was the Gaussian Naive Bayes Model as it had an highest accuracy of 76.17%, a perfect precision of 100.00%, and a weak 50.69% recall. Moreover, the model which performed the worst was the Decision Tree Model as it had an accuracy of only 48.67%, a precision of 48.49% and a F1 score of 65.32%.

4.5 Feature 5 - Unique Words Contained in the Positive/Negative Lexicon Database

4.5.1 Why We Chose This Text Analytic Approach

For our fifth feature we chose to analyze the unique words which appear in each document that are in the opinion lexicon from the NLTK. This approach focuses on words that are strong markers of emotion

by filtering out words in each document that exist in the positive and negative lexicons, respectively. That said, by filtering words by positive and negative lexicon, we are able to reduce the distraction of neutral words and really get an understanding of the words that are most likely to influence sentiment classification of a document.

4.5.2 How To Use This Method

In order to use this method we first loaded the positive and negative lexicons. We then aggregated words by sentiment by filtering words in each document by positive lexicon before adding it to filtered_pos. We did the same, with the negative lexicon as we filtered the words to make sure it was in the negative lexicon before adding it to filtered_neg. Following this we counted the frequencies of words in each sentiment and then selected the top 1000 words from each sentiment. Once we selected the top 1000 words from each sentiment we were able to define a function to create binary features for each document. Our final step in executing this approach was to apply the function to create features and convert the series of dictionaries to a dataframe.

4.5.3 Results

All in all, our result from this approach was a dataframe containing all the movie reviews where each column was a word that was determined to be a positive or negative word according to the opinion lexicon which we filtered for. In the columns a '1' in the column meant that the corresponding word was found in the document, and '0' meant it was not. That said, with the sentiment of each review being the last column, we were able to understand how different words in an opinion lexicon are spread out across positive and negative reviews.

4.5.4 Model Evaluation

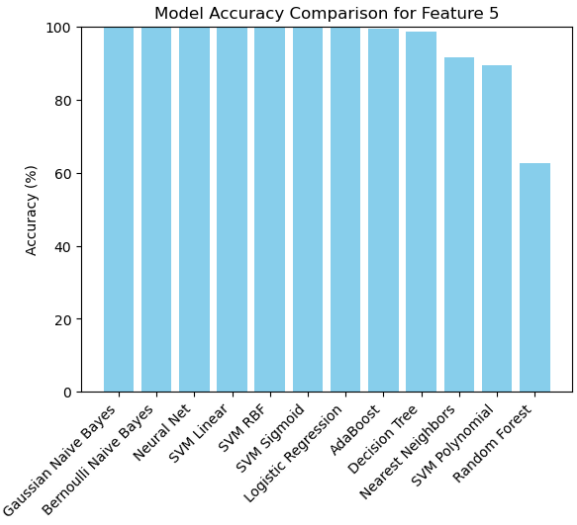

Figure 5: Model accuracy comparison for feature 5

Upon evaluating each model for this feature, we found that many of the machine learning models performed very well. Observing the plot we can notice how the Guassian Naive Bayes, Bernoulli Naive Bayes, Neural Network, SVM Linear, SVM RBF, SVM Sigmoid, Logistic Regression, Ada Boost, Decision Tree, and Nearest Neighbors are all scoring over 90% in terms of accuracy. That said, the models which performed the best were the Guassian Naive Bayes and Bernoulli Naive Bayes models as they had perfect 100% accuracy, 100% recall, 100% precision and 100% F1 score. It is important to highlight that a 100% score in all these metrics can signal a potential problem of over-fitting, as the machine leaning models have learned "too well" our training dataset and won't necessarily perform so well in a new test set of unseen data. On the other hand, the model which performed the worst was the random forest model as it had an accuracy of 62.67%, precision of 56.42% , a recall of 100%, and a

F1 score of 72.14%. The high recall for the random forest model indicates to us that the model often classifies more instances as positive, which is not always correct.

4.6 Feature 6 - Frequency of Emotional Language In Positive and Negative Reviews

4.6.1 Why We Chose This Text Analytic Approach

For this feature, we chose to harness the power of the NRCLex library to uncover not just sentiments but also the nuanced emotions embedded within the text. Through this code, we unlock the emotional pulse of the Movie Reviews Corpus, shedding light on audience sentiments and preferences, and paving the way for a deeper understanding of the cinematic experience.

4.6.2 How To Use This Method

In order to use this method we began by defining a function to process each review in the data frame using the NRCLex, which is a python library used to analyze text for emotional content. Following this, we applied the function to the DataFrame and then defined another function to extract the features for each emotion dictionary. Finally, we extracted the features for each review.

4.6.3 Results

All in all, our result from this approach was a dataframe containing all the movie reviews where the columns are emotions which are of the 6 primary emotions that NCRLex identifies and analyzes text for. Thus, we have the emotions such as sadness, disgust, trust, joy, anticipation, anger, as well as the emotion positive and negative and the sentiment of the review as the columns for this dataframe. Each row represents the frequency or intensity of the corresponding emotion in the review. These frequencies are a measure of how much of the text's emotional content can be attributed to each specific emotion. For example, a value of 0.084211 in the emotion(fear) column indicates that fear is present in the review, with a frequency of approximately 8.42%.

4.6.4 Model Evaluation

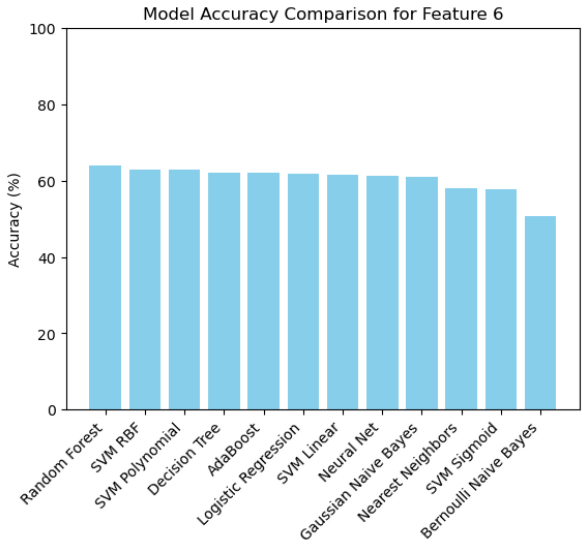

Figure 6: Model accuracy comparison for feature 6

Upon evaluating each model for this feature, the top performing model was the Random Forest, which had the highest accuracy of 63.83%. Random Forest had a precision of 64.62%, recall of 60.07%, and F1 Score of 62.26%. Nevertheless, although Random Forest had the highest accuracy, many of the

models came very close to that accuracy score such as SVM RBF, SVM Polynomial, Decision Tree, AdaBoost, Logistic Regression, etc. Moreover, the model which performed the worst is the Bernoulli Naive Bayes as it had the lowest accuracy, recall, and F1 score among all the models listed. The model had an accuracy of 50.67%, precision of 62.50%, recall of 1.68%, and F1 Score of 3.27%.

4.7 Feature 7 - Topic Models

4.7.1 Why We Chose This Text Analytic Approach

For this feature we chose to conduct a comprehensive analysis of movie reviews by leveraging LDA for topic modeling. The code integrates sentiment labels with the topic distributions, enabling a joint exploration of thematic content and sentiment expression in the reviews. It assesses the relationship between topics and sentiment, providing valuable insights into the underlying patterns and trends within the movie reviews corpus.

4.7.2 How To Use This Method

In order to use this method we began by creating a dictionary representation of the documents and then filtered out extremes to limit the number of features. We then converted the dictionary to a Bag of Words corpus for reference and then set LDA model parameters. With the LDA model parameters set we used the LDA model to get topics for each document and then applied the function to convert documents to topic distributions. Our final steps in using this method entailed converting the list of topic distributions to a DataFrame and then adding the sentiment column from the original data frame.

4.7.3 Results

All in all, our result from this approach was a dataframe containing all the movie reviews where the columns were listed 0 to 9 and the last column had the sentiment. The columns 0-9 correspond to one of the 10 topics identified by the LDA model.While each cell has a number which represents the proportion of each document that corresponds to one of the 10 topics identified by the LDA model. As per the results, some documents have more values in one or more columns while others are more evenly spread out across multiple topics.

4.7.4 Model Evaluation

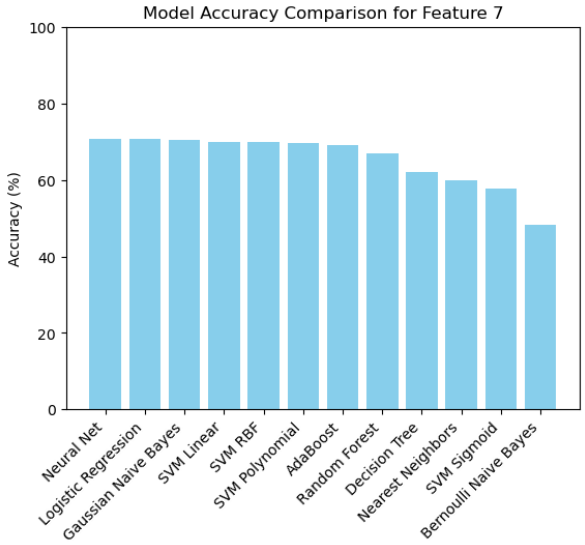

Figure 7: Model accuracy comparison for feature 7

Upon evaluating each model for this feature, we found that most of the models had very similar performance. Neural Network (MLPClassifier) was the top performer with an accuracy of 70.67%, precision of 67.70%, recall of 75.17%, and F1 Score of 71.24%. While Neural Net was the top performing model, Gaussian Naive Bayes, Random Forest, SVM Linear, SVM RBF, and SVM Polynomial had similar performances as their accuracy was only roughly 1% less than Neural Net. Moreover, the model which had the worst performance was Bernoulli Naive Bayes as it has the lowest accuracy of 48.33%, precision of 48.33%, recall of 100.00%, and F1 Score of 65.17%

4.8 Feature 8 - TF-IDF Vectors

4.8.1 Why We Chose This Text Analytic Approach

For this feature, we chose to leverage the TF-IDF Vectorizer to transform the processed text into a TF-IDF matrix. We set the maximum number of features because this method is a great way of capturing the importance of terms while filtering out noise. One of the advantages of utilizing TF-IDF is it contributes to a more thorough understanding of the text's semantic meaning and significance. This approach offers a versatility in adjusting the numbers of features while addressing a range of dataset sizes and complexities.

4.8.2 How To Use This Method

In order to use this method, we initialized the TF-IDF Vectorizer and adjusted the max_features to select the number of terms. We then fit and transformed the processed text to a TF-IDF matrix and then converted this matrix into a dataframe. Our last step to fulfill this method and create a dataframe was to add the sentiment column from the original DataFrame such that we can see the sentiment of each review in the resulting dataframe.

4.8.3 Results

All in all, our result from this approach was a dataframe containing all the movie reviews where each column in the data frame corresponds to the 2000 most important words based on the TF-IDF scores. The rows contain the TF-IDF score for those words in that particular review. The higher the value, the more unique and relevant the word is in the specific review compared to all the other reviews. Additionally, a value of zero means the word does not appear in that review or that it appears in all the reviews.

4.8.4 Model Evaluation

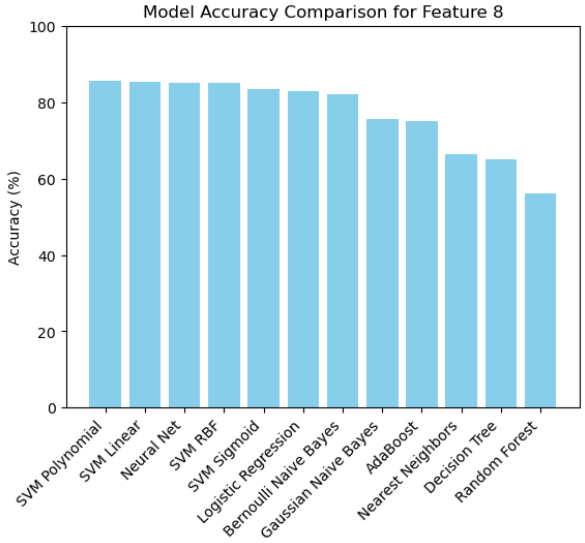

Figure 8: Model accuracy comparison for feature 8

Upon evaluating each model for this feature, we found that Support Vector Machine with a polynomial kernel emerges as the top-performing model. However, Neural Network, SVM Linear, and SVM RBF followed very closely and were only roughly 1 percent less in accuracy than SVM Polynomial. As the top performer, SVM Polynomial has an accuracy of 85.67%, precision of 85.17%, recall of 85.17%, and F1 Score of 85.17%. The model which performed the worst is random forest as it had a low accuracy of 56.00%,precision of 60.15%, recall of 61.06%, F1 Score of 57.28%

5 Conclusion

Model Accuracy Across Features

Model	Feature 1	Feature 2	Feature 3	Feature 4	Feature 5	Feature 6	Feature 7	Feature 8
Nearest Neighbors (KNN)	54.83	56.83	55.67	60.00	91.67	58.00	59.83	66.33
Decision Tree	64.00	53.67	48.83	48.67	98.67	62.00	62.17	65.17
Random Forest	56.00	55.33	48.50	50.50	62.67	63.83	66.83	56.00
Neural Network	81.67	73.33	61.17	73.67	99.83	61.33	70.67	85.17
AdaBoost	76.00	61.50	54.17	58.33	99.50	62.00	69.00	75.00
SVM Linear	80.67	69.17	60.67	67.00	99.83	61.50	70.00	85.33
SVM RBF	81.83	74.83	61.17	67.17	99.83	63.00	70.00	85.00
SVM Sigmoid	73.50	76.33	62.33	69.50	99.83	57.67	57.67	83.50
SVM Polynomial	73.00	59.83	47.67	67.83	89.33	62.83	69.67	85.67
Gaussian Naive Bayes	74.50	70.00	64.33	76.17	100.00	61.00	70.50	75.50
Bernoulli Naive Bayes	82.50	76.17	64.83	74.50	100.00	50.67	48.33	82.17
Logistic Regression	81.83	76.00	61.33	74.17	99.83	61.67	70.67	82.83

Figure 9: Model accuracy across features

In summation, by defining 8 features for extraction and testing them on all of our machine learning models, we were able to observe the difference in language used in positive/negative reviews, as well as gain insight into model performance across multiple features. In our analysis, we found that across the various features, the performance of each model varied. This is demonstrated as we saw some models such as the Gaussian Naive Bayes and Bernoulli Naive Bayes models perform very well for some features but then not perform well in other features. Adding to this, we also observed how the Decision Tree and Random Forest (except for feature 6) consistently had a poor performance across all features. Moreover, as we observed which feature had the best model performance, we found that for feature 5 in particular, all models performed very well as many of them almost reached perfect accuracy. While this sounds very good, it is important to note that there may be the issue of over-fitting given the 100% accuracy for this feature.

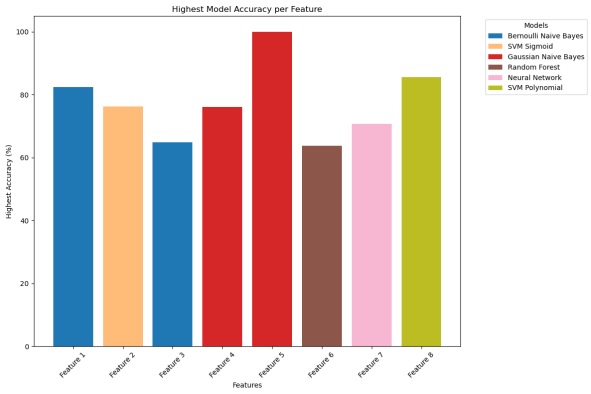

Figure 10: Bar chart of best model for accuracy across features

Furthermore, not only did we want to understand the feature where the models performed the best, but we also wanted to know the best model across all features. To do this we created the bar chart on the top where we found that Naive Bayes and Support Vector Machines were among the top-performing models across all features. Overall, having this understanding of the model performance on our features was important as it assisted us in making informed decisions for model selection as well as determining potential areas for further refinement.

6 Practical implications of examining this research question

Having conducted our analysis for our research question, it can be understood that there are a number of practical implications. For instance, our analysis will allow for improved customer insights as it will provide deeper understanding of audience preferences and perceptions. This will in turn allow industry experts to tailor content to meet audience demands. Moreover, by analyzing customer feedback and reactions, producers can identify successful themes and storytelling techniques for future productions. Additionally, market research and competitor analysis facilitated by these tools will allow stakeholders to identify emerging trends and assess competitive positioning in the industry. Finally, with this understanding of customer preferences, industry experts can enhance user experiences on streaming platforms by delivering personalized recommendations.