AIT 526

Project Report

Sentiment Analysis on Rotten Tomatoes Movie Reviews

Team 2:

1. Bryan Vega
2. Srividya Panchagnula
3. Terry Hill
4. Zhongjia Liao (Tony)

Dr. Maryam Heidari

10 May 2023

Contents

Abstract

This project involves conducting sentiment analysis on movie reviews from Rotten Tomatoes, one of the most popular movie review websites. Sentiment analysis is a natural language processing technique that involves identifying the polarity of text as positive, negative, or neutral. By analyzing the sentiment of movie reviews, we aim to gain insights into people's opinions and emotions towards movies and identify trends in how different movies are perceived.

The project follows a step-by-step process that involves collecting and cleaning the dataset, preprocessing the data, and exploring the dataset to gain insights into the distribution of reviews and the sentiment towards different movies. We use various machine learning algorithms to train and test our sentiment analysis model and evaluate its performance using standard evaluation metrics.

Our sentiment analysis model provides valuable insights into the overall sentiment of the reviews and identifies trends in how different movies are perceived. We analyze the sentiment of reviews toward different genres, actors, directors, and production houses to identify patterns and trends. We also compare the sentiment of reviews from different sources, such as critics and audiences, to gain insights into how different groups perceive movies.

Overall, this project demonstrates the usefulness of sentiment analysis in understanding people's opinions and emotions toward movies. The insights gained from this analysis can be used by filmmakers, movie studios, and marketers to understand their audience better and improve their future projects.

Keywords: Rotten Tomatoes, Sentiment Analysis, Machine Learning, TF-IDF, Polarity Classification

1. Introduction

1. Background

Sentiment analysis has become an increasingly popular tool for analyzing people's opinions and emotions about various topics, including movies. Sentiment analysis involves natural language processing (NLP) techniques to automatically identify text polarity as positive, negative, or neutral. In this project, we will conduct sentiment analysis on movie reviews from Rotten Tomatoes, one of the most popular movie review websites. Rotten Tomatoes is an online platform aggregating movie reviews from professional critics and audiences. With millions of reviews on the platform, Rotten Tomatoes is an ideal dataset for conducting sentiment analysis on movie reviews.

Movie reviews are an essential source of information for moviegoers and filmmakers alike. Reviews can influence people's decisions on which movies to watch and provide valuable feedback for filmmakers to improve their future projects. However, manually analyzing many reviews can be time-consuming and subjective. This is where sentiment analysis comes in, as it can automatically analyze the sentiment of reviews and provide insights into people's opinions and emotions toward movies.

By analyzing the sentiment of movie reviews on Rotten Tomatoes, we aim to gain insights into the overall sentiment of the reviews and identify trends in how different movies are perceived. This analysis can be helpful to movie studios and marketers to understand their audience better and improve their marketing strategies. Filmmakers can also use this analysis to gain feedback on their projects and identify areas for improvement.

Overall, sentiment analysis is valuable for analyzing movie reviews and gaining insights into people's opinions and emotions toward movies. Using natural language processing techniques, we can automatically analyze large amounts of text and provide valuable insights that inform decision-making in the movie industry.

2. Related work

There have been several related works and projects on conducting sentiment analysis on movie reviews from critics and audiences. Here are a few examples:

- Sentiment Analysis of Movie Reviews using Machine Learning Techniques: This project used machine learning algorithms to classify movie reviews as either positive or negative based on the text of the review. The study analyzed a dataset of movie reviews from IMDB, and achieved an accuracy of over 80% in classifying reviews. Bhosale, M., & Garje, G. (2017).
- Sentiment Analysis of Movie Reviews using Natural Language Processing: This project utilized natural language processing techniques to analyze movie reviews from Rotten Tomatoes and classify them as either positive, negative, or neutral. The study achieved an accuracy of 84% in classifying reviews. Agarwal, A., & Pandey, P. (2016).
- Analysis of Movie Reviews from Critics and Audiences: This project analyzed movie reviews from critics and audiences using sentiment analysis techniques. The study compared

the sentiment of the reviews from the two groups and identified differences in how they perceived movies. Itani, O. S., & Touma, H. (2018).

2. Dataset Acquisition

2.1. Overview

The Rotten Tomatoes Movies and Critic Reviews Dataset is a collection of data on movies and their corresponding critic reviews from the Rotten Tomatoes website. It contains information on 17,790 movies and 544,296 critic reviews from 2000 to 2021. The dataset size is over 1 million rows of data in CSV format.

The dataset includes details such as movie title, genre, release date, box office gross, runtime, and rating on a scale of 0 to 10. It also includes critic review data, including the critic's name, publication, review text, and review score on a scale of 0 to 100.

The dataset is available in CSV format from Kaggle and was created by scraping data from the Rotten Tomatoes website. It can be used for various purposes, such as sentiment analysis, natural language processing, and machine learning applications in the movie industry.

2.2. Dataset Attributes

Below is a brief explanation of the columns and their data types in the Rotten Tomatoes Movies and Critic Reviews Dataset:

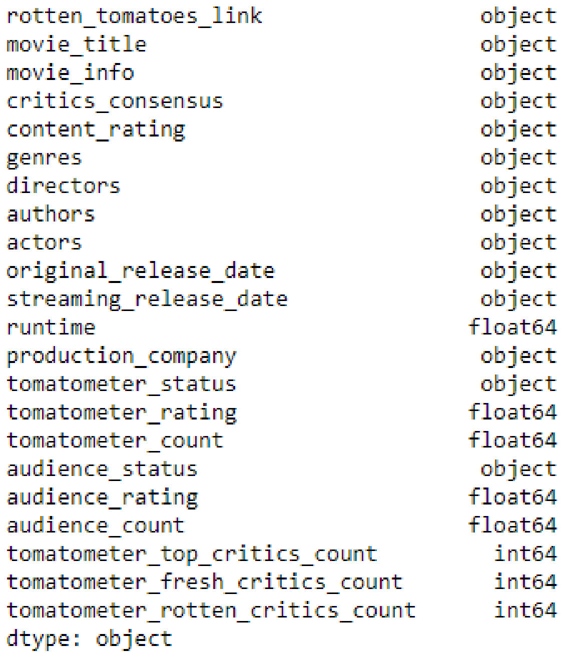
The Rotten Tomatoes Movies Dataset consists of the following fields:

- rotten_tomatoes_link – URL link to the movie for the row. Navigate to this by placing https://www.rottentomatoes.com/ in front of the field data
- movie_title – Title of the movie
- movie_info – Brief description of the movie
- critics_consensus - Summary of consensus among critics about the movie
- content_rating – Classification of the movie based on the target audience and type of explicit content
- genres – List the genre of the film that the movie belongs to, such as action or fantasy
- Directors - Names of person or persons who directed the movie
- Authors – Name of person or persons who wrote the screenplay for the movie
- Actors – Names of the main actors and actresses featured in the movie
- original_release_date – Date when the movie was first released in theaters
- streaming_release_date – The date when the movie became available via streaming platforms
- Runtime – Duration of the movie in minutes
- production_company – Name of company or companies that produced the movie
- tomatometer_status – Rotten tomatoes categorical assessment of the movie’s reception by critics
- tomatometer_rating – Percentatge of positive reviews by critics, from 0 to 100
- tomatometer_count – Total number of critic reviews used to calculate the Tomatometer rating

- audience_status - Categorical assessment of the movie's reception by the general audience, such as "Spilled," "Upright," or "Tipped"
- audience_rating - Average rating from the general audience, usually on a scale of 1 to 5, with 5 being the highest
- audience_count - Total number of audience members who have rated the movie
- tomatometer_top_critics_count - Number of reviews from top critics used to calculate the Tomatometer rating
- tomatometer_fresh_critics_count - Number of positive reviews from critics used to calculate the Tomatometer rating
- tomatometer_rotten_critics_count - Number of negative reviews from critics used to calculate the Tomatometer rating

The following figure shows the data types of each field within their respective dataset.

rotten_tomatoes_movies.csv



```
rotten_tomatoes_link                  object
movie_title                           object
movie_info                            object
critics_consensus                     object
content_rating                        object
genres                                object
directors                             object
authors                               object
actors                                object
original_release_date                 object
streaming_release_date                object
runtime                              float64
production_company                    object
tomatometer_status                    object
tomatometer_rating                   float64
tomatometer_count                    float64
audience_status                       object
audience_rating                      float64
audience_count                       float64
tomatometer_top_critics_count          int64
tomatometer_fresh_critics_count        int64
tomatometer_rotten_critics_count       int64
dtype: object
```

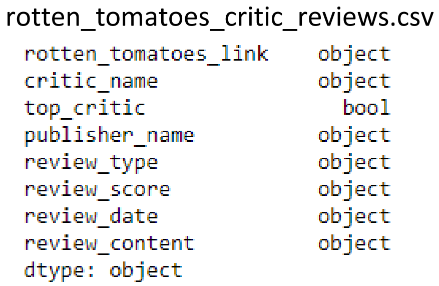
Figure 1: rotten_tomatoes_movies.csv object type

The Rotten Tomatoes Critic Reviews Dataset consists of the following fields:

- Rotten_tomatoes_link: The URL of the movie page on the Rotten Tomatoes website.
- Critic_name: The name of the critic who wrote the review.
- Top_critic: A binary flag indicating whether the critic is a "top critic" on Rotten Tomatoes. The website's editorial team chooses top critics based on their quality and influence in the industry.
- Publisher_name: The name of the publication or website that the critic works for.

- Review_type: A categorical variable indicating the type of the review, such as "Fresh" or "Rotten".
- Review_score: A numerical score assigned by the critic to the movie being reviewed. The score is usually out of 100.
- Review_date: The date on which the review was published.
- Review_content: The text provides a more detailed evaluation of the movie.

The following figure shows the data types of each field within their respective dataset.



```
rotten_tomatoes_critic_reviews.csv
  rotten_tomatoes_link    object
  critic_name             object
  top_critic                bool
  publisher_name          object
  review_type             object
  review_score            object
  review_date             object
  review_content          object
  dtype: object
```

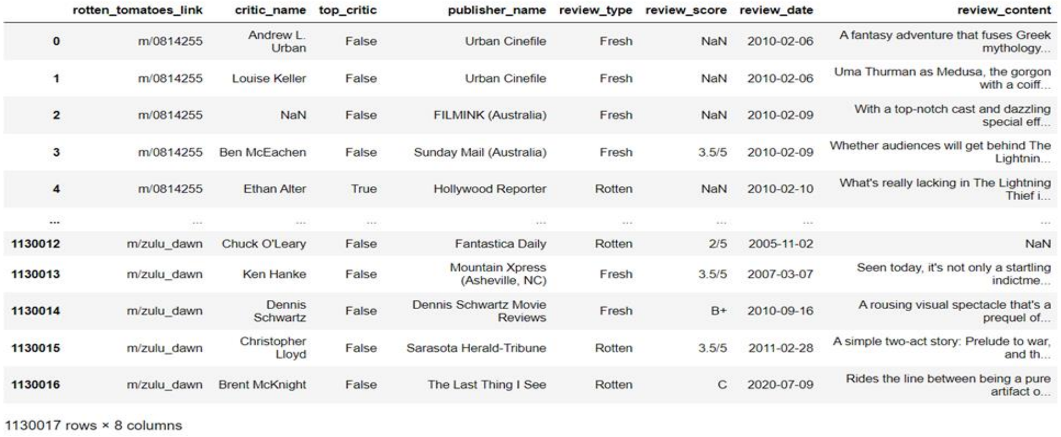
Figure 2: rotten_tomatoes_critic_reviews.csv object type

2.3. Data preprocessing

Data cleaning and preprocessing are essential to detect and correct corrupt or inaccurate records from the original data to ensure the data is appropriate to build the model. The data preprocessing procedures include providing a statistical summary of the data, identifying and removing null values, and converting the review content column to lowercase. Typical NLP text preprocessing, including removing special characters, punctuation, spaces, and stop words, will be incorporated into our final report.

2.4. Data Overview

The movie review data is a CSV file that originated from Kaggle.com. This dataset was imported into Jupyter Notebook for initial data preprocessing. The following graph shows the raw data frame in a table format.



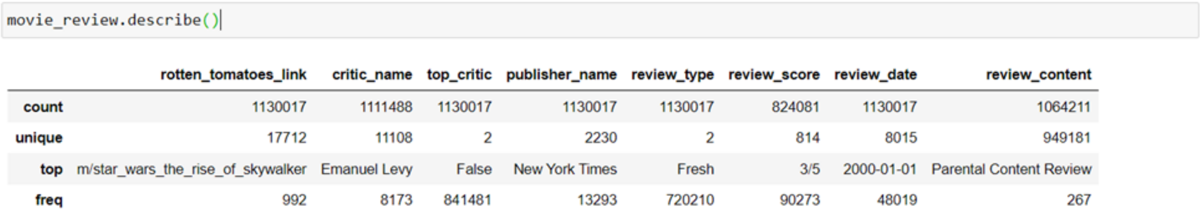
	rotten_tomatoes_link	critic_name	top_critic	publisher_name	review_type	review_score	review_date	review_content
0	m/0814255	Andrew L. Urban	False	Urban Cinefile	Fresh	NaN	2010-02-06	A fantasy adventure that fuses Greek mythology...
1	m/0814255	Louise Keller	False	Urban Cinefile	Fresh	NaN	2010-02-06	Uma Thurman as Medusa, the gorgon with a coiff...
2	m/0814255	NaN	False	FILMINK (Australia)	Fresh	NaN	2010-02-09	With a top-notch cast and dazzling special eff...
3	m/0814255	Ben McEachen	False	Sunday Mail (Australia)	Fresh	3.5/5	2010-02-09	Whether audiences will get behind The Lightnin...
4	m/0814255	Ethan Alter	True	Hollywood Reporter	Rotten	NaN	2010-02-10	What's really lacking in The Lightning Thief i...
...	...	...	...	...	...	...	...	...
1130012	m/zulu_dawn	Chuck O'Leary	False	Fantastica Daily	Rotten	2/5	2005-11-02	NaN
1130013	m/zulu_dawn	Ken Hanke	False	Mountain Xpress (Asheville, NC)	Fresh	3.5/5	2007-03-07	Seen today, it's not only a startling indictme...
1130014	m/zulu_dawn	Dennis Schwartz	False	Dennis Schwartz Movie Reviews	Fresh	B+	2010-09-16	A rousing visual spectacle that's a prequel of...
1130015	m/zulu_dawn	Christopher Lloyd	False	Sarasota Herald-Tribune	Rotten	3.5/5	2011-02-28	A simple two-act story: Prelude to war, and th...
1130016	m/zulu_dawn	Brent McKnight	False	The Last Thing I See	Rotten	C	2020-07-09	Rides the line between being a pure artifact o...

1130017 rows × 8 columns

Figure 3: Dataset objects

2.5. Statistical Summary

The below statistical description table provides an overview of the dataset including the number count, the unique value of each column, the top value and the frequency of the value.



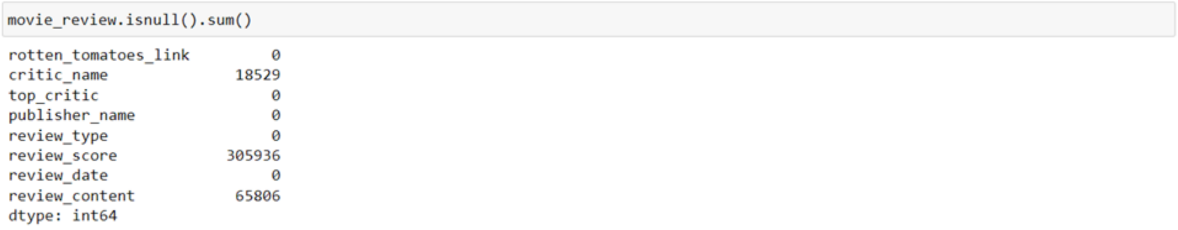
```
movie_review.describe()
```

	rotten_tomatoes_link	critic_name	top_critic	publisher_name	review_type	review_score	review_date	review_content
count	1130017	1111488	1130017	1130017	1130017	824081	1130017	1064211
unique	17712	11108	2	2230	2	814	8015	949181
top	m/star_wars_the_rise_of_skywalker	Emanuel Levy	False	New York Times	Fresh	3/5	2000-01-01	Parental Content Review
freq	992	8173	841481	13293	720210	90273	48019	267

Figure 4: Description of movie_review dataset

2.6. Null Value Summary

Null values will negatively affect our data model performance and will cause data accuracy issues in our analysis. The following summary table reveals how many records contain null values in the data frame and mostly fall under columns ‘critic_name’, ‘review_score’, and ‘review_content’.



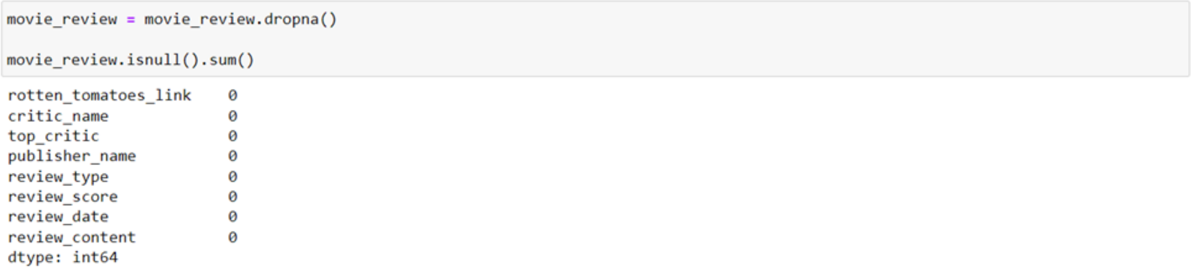
```
movie_review.isnull().sum()
```

```
rotten_tomatoes_link         0
critic_name              18529
top_critic                   0
publisher_name               0
review_type                  0
review_score            305936
review_date                  0
review_content           65806
dtype: int64
```

Figure 5: Summary of null values

2.7. Handling missing values

By implementing the Python dropna() function, we were able to handle and remove those missing values identified above. The below screenshot indicates that all missing values were handled.



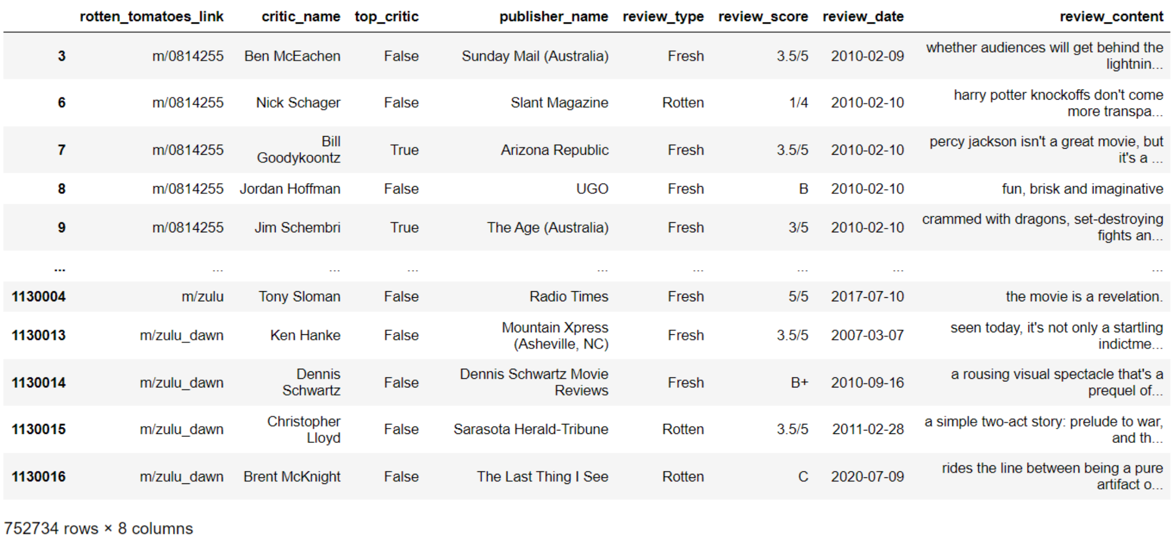
```
movie_review = movie_review.dropna()

movie_review.isnull().sum()
```

```
rotten_tomatoes_link    0
critic_name             0
top_critic              0
publisher_name          0
review_type             0
review_score            0
review_date             0
review_content          0
dtype: int64
```

Figure 7: Updated summary of null values

2.8. Lower-case conversion

Lowercase conversion is done to standardize the text and ensure that the algorithm doesn't treat the same word differently due to differences in capitalization.



	rotten_tomatoes_link	critic_name	top_critic	publisher_name	review_type	review_score	review_date	review_content
3	m/0814255	Ben McEachen	False	Sunday Mail (Australia)	Fresh	3.5/5	2010-02-09	whether audiences will get behind the lightnin...
6	m/0814255	Nick Schager	False	Slant Magazine	Rotten	1/4	2010-02-10	harry potter knockoffs don't come more transpa...
7	m/0814255	Bill Goodykoontz	True	Arizona Republic	Fresh	3.5/5	2010-02-10	percy jackson isn't a great movie, but it's a ...
8	m/0814255	Jordan Hoffman	False	UGO	Fresh	B	2010-02-10	fun, brisk and imaginative
9	m/0814255	Jim Schembri	True	The Age (Australia)	Fresh	3/5	2010-02-10	crammed with dragons, set-destroying fights an...
...	...	...	...	...	...	...	...	...
1130004	m/zulu	Tony Sloman	False	Radio Times	Fresh	5/5	2017-07-10	the movie is a revelation.
1130013	m/zulu_dawn	Ken Hanke	False	Mountain Xpress (Asheville, NC)	Fresh	3.5/5	2007-03-07	seen today, it's not only a startling indictme...
1130014	m/zulu_dawn	Dennis Schwartz	False	Dennis Schwartz Movie Reviews	Fresh	B+	2010-09-16	a rousing visual spectacle that's a prequel of...
1130015	m/zulu_dawn	Christopher Lloyd	False	Sarasota Herald-Tribune	Rotten	3.5/5	2011-02-28	a simple two-act story: prelude to war, and th...
1130016	m/zulu_dawn	Brent McKnight	False	The Last Thing I See	Rotten	C	2020-07-09	rides the line between being a pure artifact o...

752734 rows × 8 columns

Figure 8: Data frame after converting review_content to lower case

2.9. Text Preprocessing

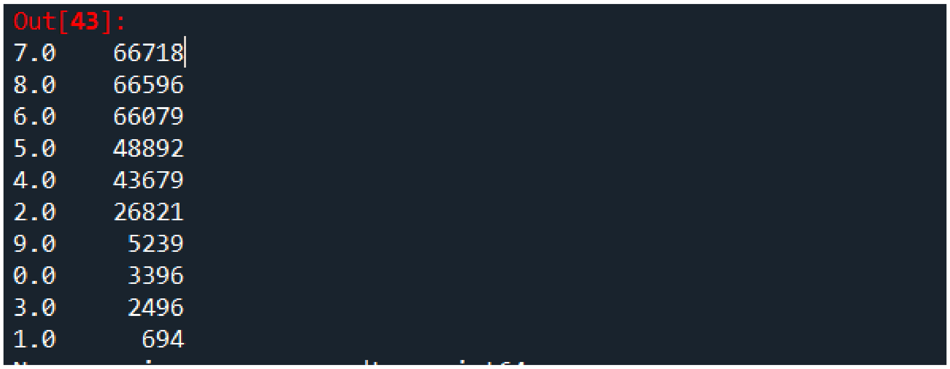
Text preprocessing is an important step in sentiment analysis that involves cleaning and transforming the raw text data into a format that can be used by machine learning algorithms. In this project, we will be conducting sentiment analysis on movie reviews from Rotten Tomatoes. Involved in text preprocessing for this project:

- Text Normalization: The text normalization step involves converting the text to a consistent format by applying various techniques like removing punctuation marks, and expanding contractions. We will also remove any numbers and special characters (including emoticons).
- Stopword Removal: Stopwords are common words that do not add much meaning to the text and can be safely removed without affecting the overall sentiment of the text. Examples of stopwords include "the," "and," "a," etc. We will remove these stopwords from the text data to reduce the number of unique words and improve the accuracy of the sentiment analysis.
- Stemming or Lemmatization: Stemming or lemmatization involves reducing words to their base form to reduce the number of unique words and improve the accuracy of the sentiment analysis. We will use the Porter stemmer algorithm to stem the words.
- Tokenization: Tokenization involves splitting the text into individual words, which can be used as features by the machine learning algorithms. We will use a tokenizer to split the text into individual words, representing a bag of words model.

- Vectorization: The final step is to convert the text data into a numerical format that the machine learning algorithms can use. We will use the term frequency-inverse document frequency (TF-IDF) algorithm to convert the text into a vectorized format.

2.10. Exploratory Data Analysis

Figure 9 below shows the review score's number counts to provide some basic understanding of the dataset. Based on this output, we were able to conclude that the majority of the movie reviews have a score of 5 or higher.

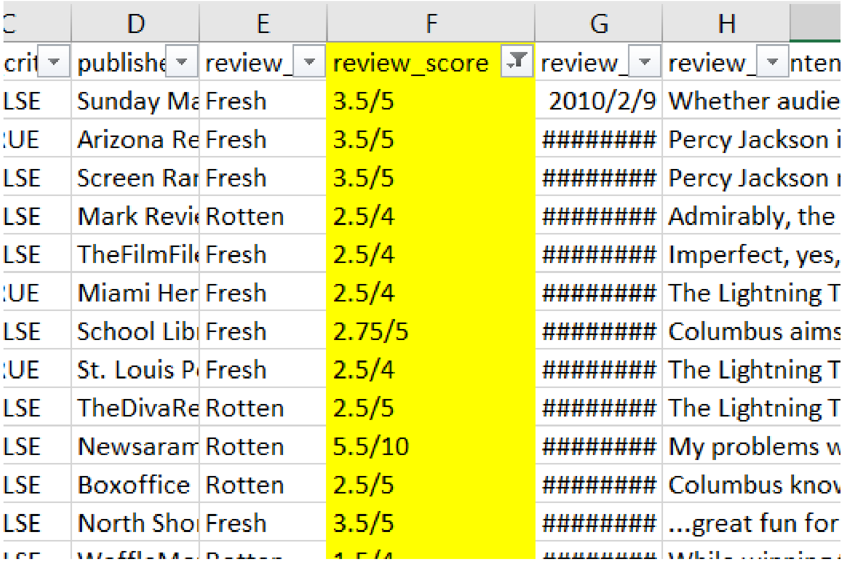


```
Out[43]:
7.0     66718
8.0     66596
6.0     66079
5.0     48892
4.0     43679
2.0     26821
9.0      5239
0.0      3396
3.0      2496
1.0       694
```

Figure 9: Review_score counts

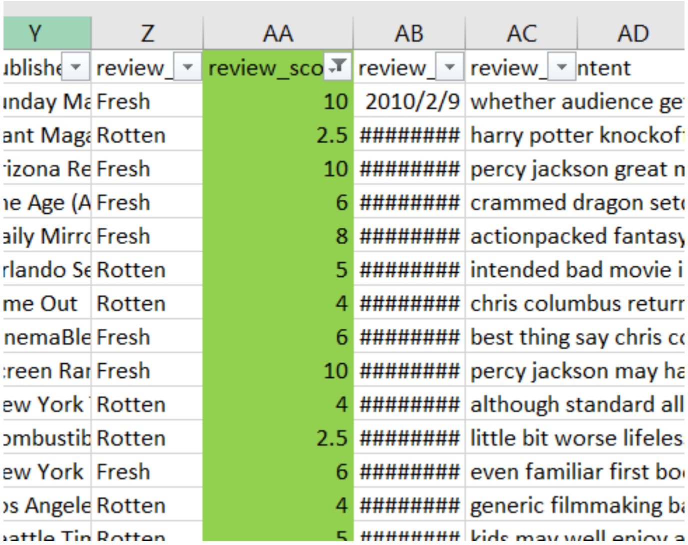
2.11. Feature Engineering

The feature engineering technique is being used to process and transform raw data to a variable that is easy to use and/or calculate in the model. The below screen shows the original dataset with different review score scale on a single column. By implementing the feature engineering, we were able to transform/consolidate the review score column into a scale of 10 rating.



C	D	E	F	G	H	
crit	publishe	review_	review_score	review_	review_	nten
LSE	Sunday Ma	Fresh	3.5/5	2010/2/9	Whether audie	
UE	Arizona Re	Fresh	3.5/5	########	Percy Jackson i	
LSE	Screen Rar	Fresh	3.5/5	########	Percy Jackson i	
LSE	Mark Revie	Rotten	2.5/4	########	Admirably, the	
LSE	TheFilmFile	Fresh	2.5/4	########	Imperfect, yes,	
UE	Miami Her	Fresh	2.5/4	########	The Lightning T	
LSE	School Libi	Fresh	2.75/5	########	Columbus aims	
UE	St. Louis P	Fresh	2.5/4	########	The Lightning T	
LSE	TheDivaRe	Rotten	2.5/5	########	The Lightning T	
LSE	Newsaram	Rotten	5.5/10	########	My problems w	
LSE	Boxoffice	Rotten	2.5/5	########	Columbus knov	
LSE	North Sho	Fresh	3.5/5	########	...great fun for	

Figure 10: Original dataset



Y	Z	AA	AB	AC	AD
ıblishe ▾	review_ ▾	review_sco ▾	review_ ▾	review_ ▾	ntent
ınday Ma	Fresh	10	2010/2/9	whether audience ge	
ant Maga	Rotten	2.5	########	harry potter knockof	
izona Re	Fresh	10	########	percy jackson great n	
ıe Age (A	Fresh	6	########	crammed dragon set	
aily Mirrc	Fresh	8	########	actionpacked fantasy	
rlando Se	Rotten	5	########	intended bad movie i	
me Out	Rotten	4	########	chris columbus returr	
nemaBle	Fresh	6	########	best thing say chris cc	
reen Rar	Fresh	10	########	percy jackson may ha	
ew York	Rotten	4	########	although standard all	
ɔmbustib	Rotten	2.5	########	little bit worse lifeles	
ew York	Fresh	6	########	even familiar first bo	
ɔs Angele	Rotten	4	########	generic filmmaking ba	
ıattle Tin	Rotten	5	########	kids may well enjoy a	

Figure 11: Updated dataset

3. Analytics and Model Development

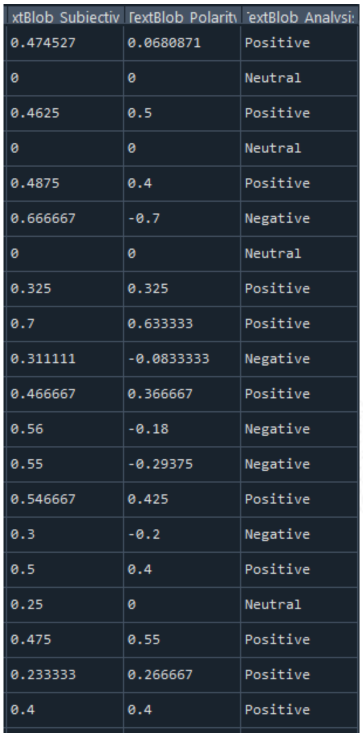
3.1. Sentiment Analysis

3.1.1. Text Blob

The NLTK package (Natural Language Toolkit) has been utilized to perform sentiment analysis for the project. NLTK is a suite of libraries and programs for symbolic and statistical natural language processing (NLP) for English written in Python. It also provides other valuable functions like WordNet, Stemming, Tagging, and Semantic reasoning to process text data.

TextBlob is another powerful Python package to process text data and analyze sentiment. We can utilize TextBlob to identify part-of-speech tagging for each word, like flagging nouns, verbs, and adjectives. The sentiment analysis function in TextBlob is also recommended to analyze text data to determine if the tone of the data is negative, neutral, or positive. It uses a machine learning algorithm trained on a large corpus of movie reviews to determine the sentiment of the text and returns a sentiment polarity score ranging from -1 (negative) to +1 (positive). TextBlob also includes a pre-trained model for part-of-speech tagging, making it easy to identify and extract different parts of speech from text.

The polarity score provided by TextBlob ranges from -1 (most negative) to 1 (most positive), with a score of 0 indicating neutral sentiment. TextBlob can also be used to identify the subjectivity of the text, providing a measure of how opinionated or objective the language used in the reviews is by analyzing the polarity and subjectivity scores of the reviews, it is possible to gain insights into the overall sentiment and tone of the reviews, as well as to identify potential biases or trends in the language used.



xtBlob_Subjectiv	TextBlob_Polarity	extBlob_Analysis
0.474527	0.0680871	Positive
0	0	Neutral
0.4625	0.5	Positive
0	0	Neutral
0.4875	0.4	Positive
0.666667	-0.7	Negative
0	0	Neutral
0.325	0.325	Positive
0.7	0.633333	Positive
0.311111	-0.0833333	Negative
0.466667	0.366667	Positive
0.56	-0.18	Negative
0.55	-0.29375	Negative
0.546667	0.425	Positive
0.3	-0.2	Negative
0.5	0.4	Positive
0.25	0	Neutral
0.475	0.55	Positive
0.233333	0.266667	Positive
0.4	0.4	Positive

Figure 12: TextBlob Analysis Output

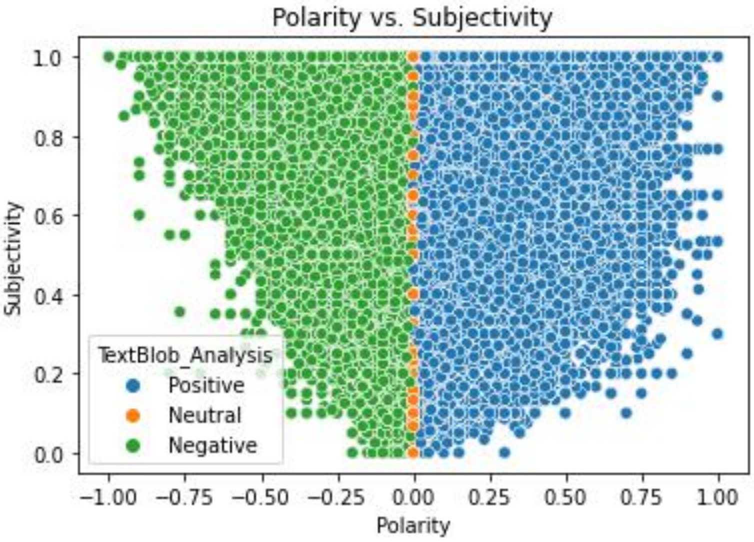


Figure 13: Polarity Visualization

Below are visualizations generated using our data to gain insight into its polarity and subjectivity.

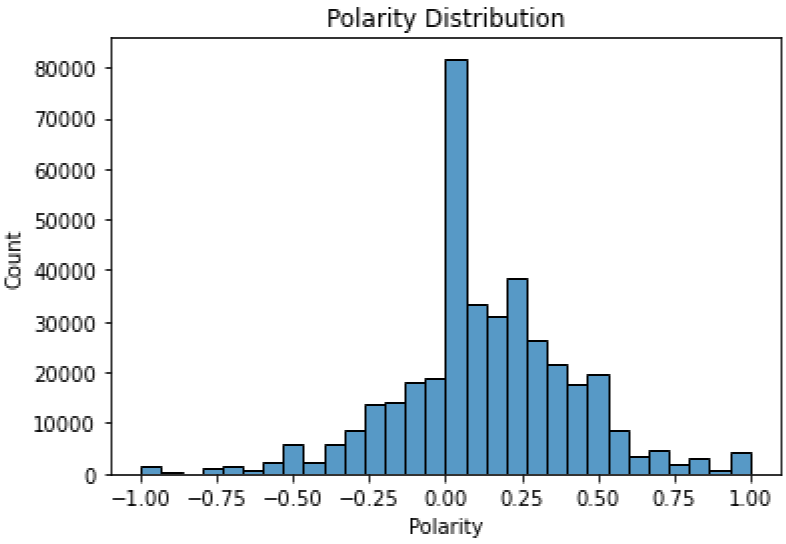


Figure 14: Polarity Distribution

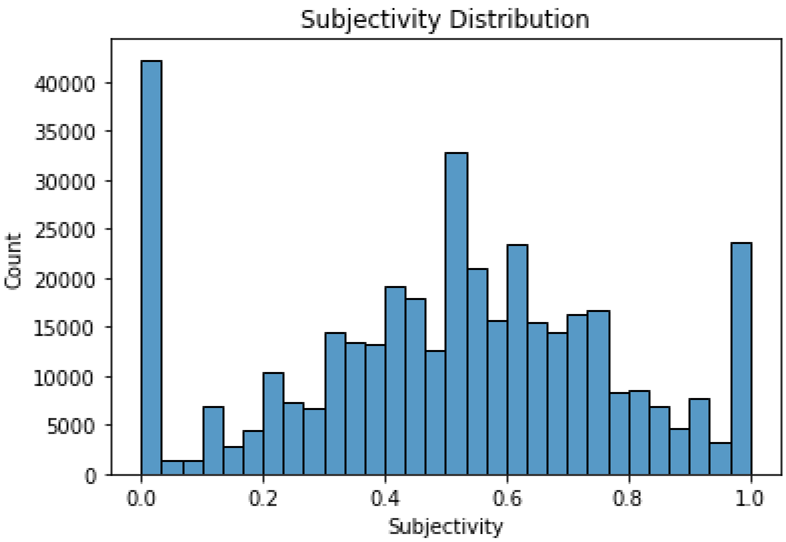


Figure 15: Subjectivity Distribution

In Sentiment distribution as Pie Chart: A pie chart displays the percentage of reviews in each category, providing a quick and easy-to-understand overview of the overall sentiment towards the movie. The size of each pie chart slice corresponds to the percentage of reviews in each category. The pie chart can be customized to include labels or colors for each category, making it easier to read and interpret.

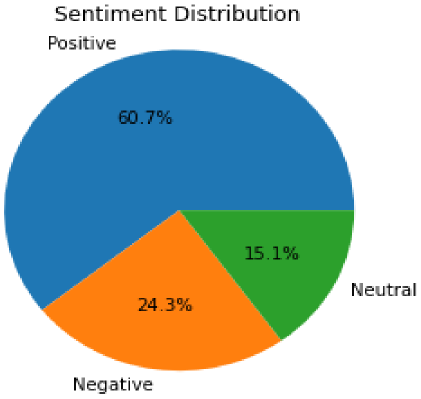


Figure 16: Sentiment Distribution as a Pie Chart

Sentiment analysis time series is a technique used to analyze changes in sentiment over time. It involves analyzing sentiment data from movie reviews over a period of time. The history of movie-making dates back to the late 19th century, known as the "silent era." The first motion pictures were produced in the 1890s, so there is no data before the 1890s. Our data is centered around and after the early 2000s for time series, and green dominance shows positive reviews for the majority.

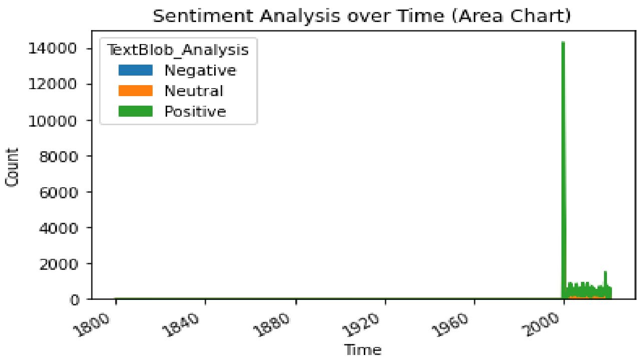


Figure 17: Sentiment Analysis over Time

3.1.2. Text Analysis

Text analysis, also known as text mining or text analytics, is extracting meaningful insights from unstructured text data. One of the most common techniques used in text analysis is identifying frequently used words in a text or corpus of texts. By identifying and analyzing the frequency of certain words, text analysis can help to reveal patterns and trends within the data, providing valuable insights into the topics and themes being discussed. This information can be used to identify patterns and trends in consumer behavior, public opinion, or any other area where text data is available.

Frequently used words can be identified using tools such as word frequency analysis, which involves counting the number of times a word appears in a text or corpus of texts. These results can then be visualized using techniques such as word clouds or bar charts, which make it easy to see which words are used most frequently. Text analysis can be applied to various text data, including social media posts, customer reviews, news articles, etc. It can be used to gain insights into customer sentiment, monitor public opinion on a given topic, or identify emerging trends and topics of discussion. In the below visualizations you can see how the data differs as it is converted from Tokens (Words) to Bigrams, and Trigrams.

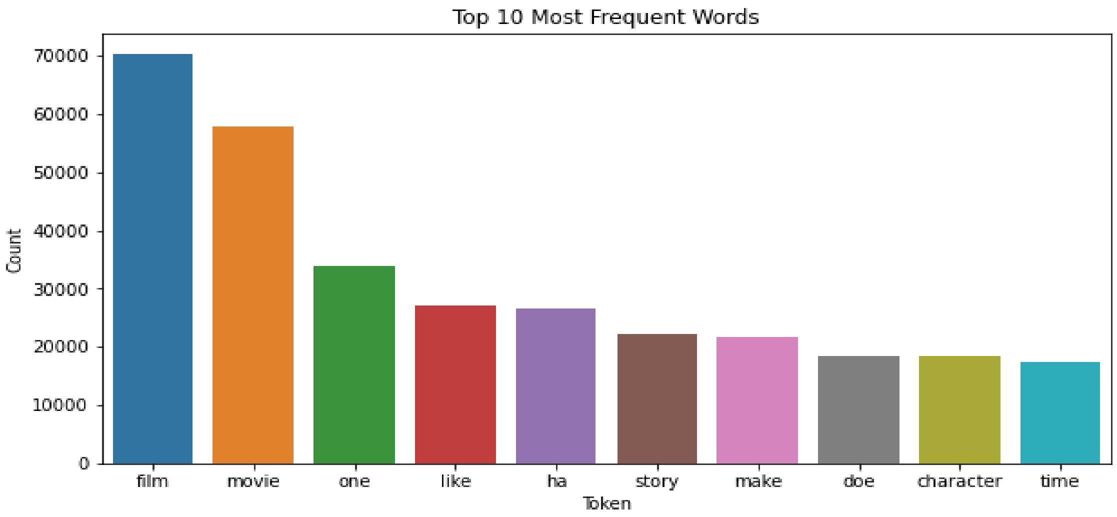


Figure 18: Top 10 Words Used

Bigrams and trigrams are powerful techniques in text analysis that can provide deep insights into the underlying structure of textual data. By identifying frequently occurring pairs or groups of words, these techniques can help reveal patterns and relationships between words and concepts. This is particularly useful in sentiment analysis, where bigram and trigram analysis can help uncover common topics and themes associated with positive or negative sentiment. Bigram and trigram analysis can also help to identify trends over time, making it a valuable tool for tracking changes in customer sentiment or opinions. Overall, bigram and trigram analysis can provide a deeper understanding of the text data and help to uncover valuable insights that might not be apparent through other forms of analysis

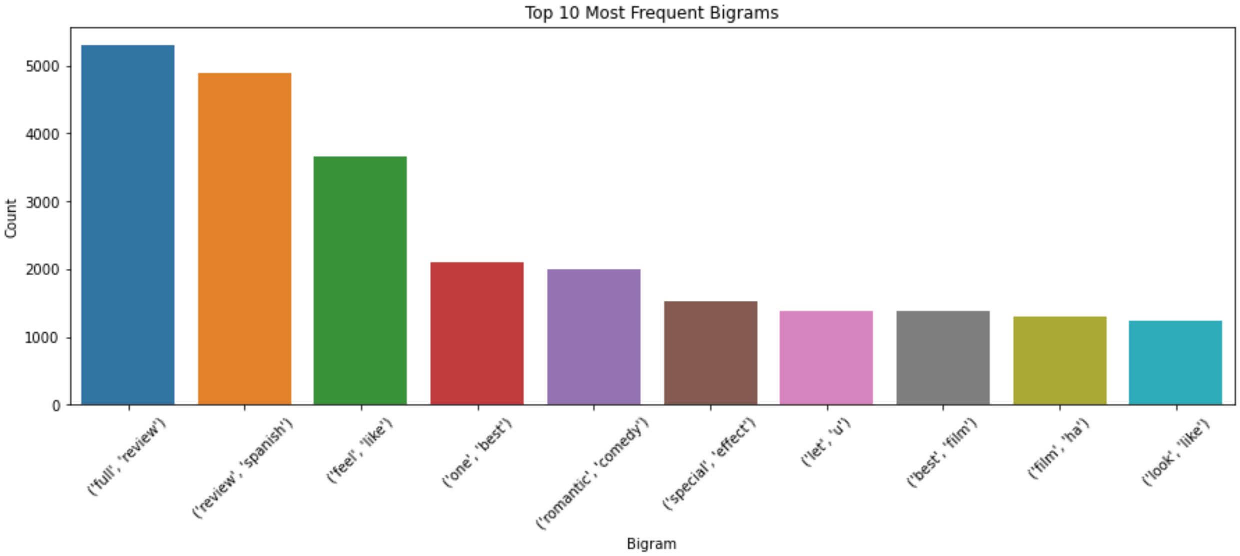


Figure 19: Top 10 Bigrams

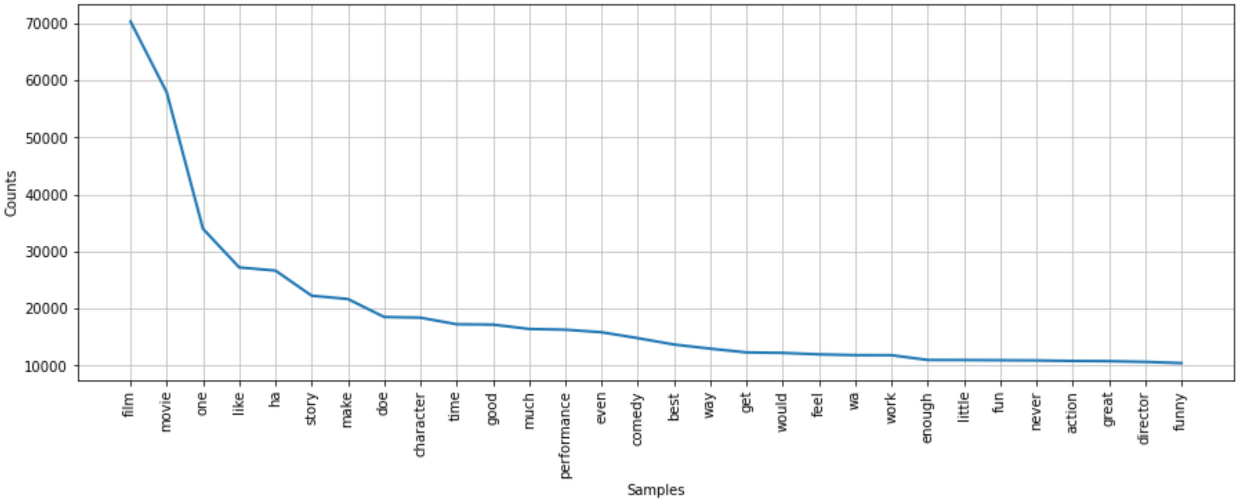


Figure 20: Top 10 Trigrams

3.1.2 Word Cloud

Word Cloud analysis is a visualization technique that allows users to view the big picture of the text data of which words have the highest frequency within the data. Word clouds are a simple yet effective visualization technique that can help to quickly identify the most commonly used words in a corpus of text. In the context of movie review sentiment analysis, word clouds can highlight the most frequently mentioned actors, directors, genres, and other key themes and topics. By comparing word clouds across different movies or genres, it is possible to identify review patterns and trends and gain insights into audience preferences. Word clouds can also identify the most commonly used positive or negative words in the reviews, providing insights into the overall sentiment toward the movie. Word clouds are a valuable tool for analyzing large amounts of text data and gaining a deeper understanding of the main themes and trends.

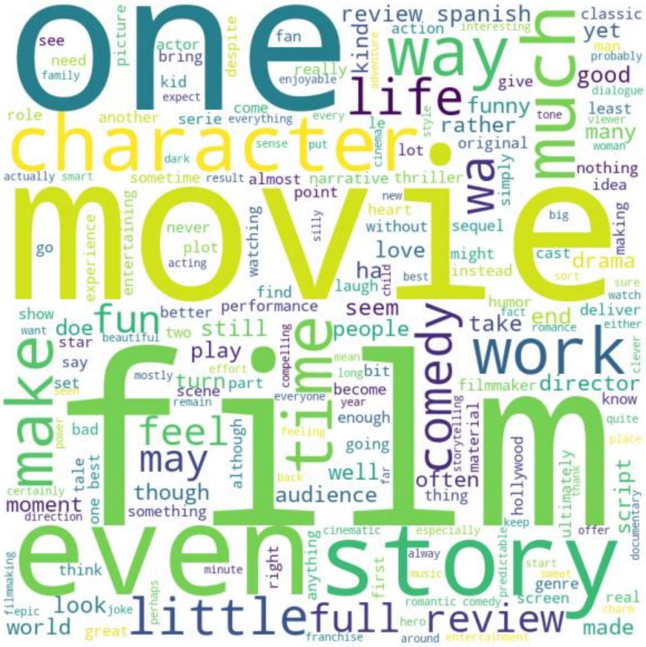


Figure 21: Word Cloud

3.1.4 Stemming

Stemming is an NLP technique to reduce the text to its base form to improve analysis and model accuracy. For example, consider converting the words “come”, “came”, and “coming” to the root word “come” after the stemming technique is utilized. This will reduce the number of the same word being input to the ML model and minimize the confusion of the words with the same meaning.

3.2. Model Development

Following preprocessing, sentiment, and text analysis, we built the model. In developing the model we split the data into a 80/20 training and testing set and used TF-IDF vectorization to transform the text data.

We used the classification algorithms Logistic Regression, Decision Tree, and k-Nearest Neighbor to evaluate the performance and visualized the results of each model using the Confusion Matrix Heatmap. Finally, the best-performed model was chosen based on accuracy.

3.2.1 Model Characteristics

The Logistic Regression model, a widely-used machine learning algorithm for binary and multi-class classification problems, analyze review contents with sentiment labels in three classes: 'Negative,' 'Neutral,' and 'Positive'. The dataset is initially split into an 80% training set and a 20% testing set to train the model. The text data from the review content are then transformed into numerical data utilizing the TF-IDF (Term Frequency-Inverse Document Frequency) vectorizer, enabling the algorithm to effectively process and analyze the text. Subsequently, a pipeline is constructed, incorporating data standardization through the StandardScaler (without mean) and training the Logistic Regression model with a maximum of 1000 iterations. The performance of the trained model is assessed on the test set using evaluation metrics such as accuracy, classification report, and confusion matrix. Visualization techniques, including a heatmap of the normalized confusion matrix and ROC (Receiver Operating Characteristic) curves for each sentiment class ('Negative,' 'Neutral,' and 'Positive'), are employed to facilitate the interpretation of the model's performance and to identify potential areas for improvement.

The Decision Tree model, another widely-utilized machine learning algorithm for classification tasks, is also applied to classify review sentiments as 'Negative,' 'Neutral,' or 'Positive.' Decision Trees employ hierarchical structures that recursively divide the data based on feature values, aiming to maximize information gain at each node. The Decision Tree model is initialized and trained using the same training set (80% of the data) as the Logistic Regression model. The text data from the reviews have already been converted into numerical data using the TF-IDF vectorizer. The trained Decision Tree model is evaluated on the test set, employing the same evaluation metrics as those used for the Logistic Regression model: accuracy, classification report, and confusion matrix. Visualizations for this model are also generated, including a heatmap of the confusion matrix and ROC curves for each sentiment class ('Negative,' 'Neutral,' and 'Positive'). These visualizations allow for comparing the model's performance with that of the Logistic Regression model and help identify potential improvement areas.

The k-Nearest Neighbors (k-NN) algorithm is another standard method used in classification problems, and it can also be applied to sentiment classification tasks such as determining if a review is 'Negative,' 'Neutral,' or 'Positive.' Unlike the Logistic Regression and Decision Tree models, k-NN is an instance-based learning algorithm. It does not create an internal model but instead bases its predictions on the data instances closest to the one being classified. Applying k-NN to sentiment analysis involves the same initial steps as the previous models: splitting the dataset into a training set (80%) and a test set (20%) and converting the text data from the reviews into numerical data using the TF-IDF vectorizer. After these steps, the k-NN algorithm can be trained using the training set.

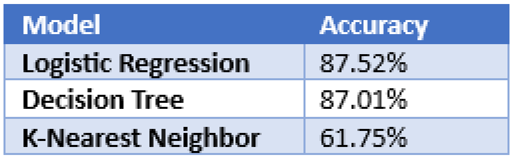
3.2.2 Model Selection

Upon evaluating the performance of the Logistic Regression, Decision Tree, and K-Nearest Neighbor models for sentiment analysis, the Logistic Regression model is determined to be the most suitable choice. This selection is based on several key factors contributing to its superior performance.

The Logistic Regression model achieves the highest accuracy of 0.875 among the three models, signifying its ability to correctly predict a greater proportion of sentiments. Furthermore, it demonstrates strong precision, recall, and F1-score performance across the 'Negative' and 'Neutral' classes, outperforming the K-Nearest Neighbor model. With a macro average F1-score of 0.84, the Logistic Regression model showcases a well-rounded performance, which is particularly valuable for handling imbalanced datasets.

Additionally, an analysis of the confusion matrices reveals that the Logistic Regression model's predictions for the 'Negative', 'Neutral', and 'Positive' classes are superior to those of the K-Nearest Neighbor model. Although the Decision Tree model exhibits relatively similar performance, the Logistic Regression model maintains a slightly better balance between true positives and false positives/negatives.

The Logistic Regression model's excellent performance across various evaluation metrics and balanced predictions across all sentiment classes establish it as the most appropriate choice for sentiment analysis in this context. This model will provide accurate and reliable sentiment predictions, making it a valuable tool for understanding and analyzing user opinions.



Model	Accuracy
Logistic Regression	87.52%
Decision Tree	87.01%
K-Nearest Neighbor	61.75%

Table 1: Model Accuracy Comparison

3.2.3 Hyperparameter Tuning

Hyperparameter tuning optimizes a model's parameters that are not learned during training. Improving these parameters can lead to significant gains in the model's performance. In this project, hyperparameter tuning is achieved using a grid search approach in conjunction with cross-validation. Following identifying hyperparameters for the Logistic Regression model through GridSearchCV, the model is retrained using the identified hyperparameters and evaluated. The Logistic Regression model's performance may be improved through hyperparameter optimization. However, it is important to note that this process could increase the code’s runtime since it entails an extensive search through multiple hyperparameter combinations.

4. Visualizations and Analysis

This section highlights key visualizations depicting the model and its findings.

4.1 Classification Matrix

The classification report comprehensively evaluates a model's performance in predicting Negative, Neutral, and Positive classes. Several key metrics assess the model's effectiveness in making accurate predictions.

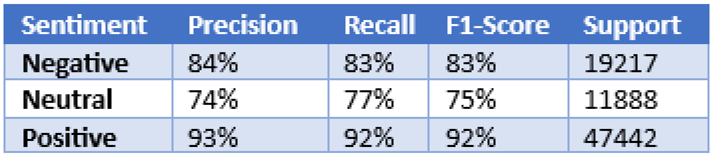
Precision, which measures the proportion of true positive predictions out of all positive predictions made by the model, helps gauge how well the model avoids false positives. For the Negative class, the model's precision is 0.84, 0.74 for the Neutral class, and an impressive 0.93 for the Positive class. This means that the model is especially proficient at identifying positive instances.

Another important metric calculates the proportion of true positive predictions out of all positive instances. A higher recall suggests fewer false negatives. In this case, the model achieves a recall of 0.83 for the Negative class, 0.77 for the Neutral class, and 0.92 for the Positive class. These values indicate that the model performs relatively well in detecting true positives across all classes.

The F1-score, a metric that balances precision and recall by computing their harmonic mean, provides an overall sense of the model's performance. The F1 scores for the Negative, Neutral, and Positive classes are 0.83, 0.75, and 0.92, respectively, further highlighting the model's strength in identifying positive instances.

The support metric reveals the number of actual instances for each class in the test dataset, which consists of 19,217 Negative, 11,888 Neutral, and 47,446 Positive instances.

The classification report demonstrates that the model performs well overall, with an accuracy of 88%. It is particularly adept at classifying positive instances, though its precision and recall for neutral instances are slightly lower. This information is valuable for understanding the model's strengths and weaknesses and guiding potential future iterations' improvements.



Sentiment	Precision	Recall	F1-Score	Support
Negative	84%	83%	83%	19217
Neutral	74%	77%	75%	11888
Positive	93%	92%	92%	47442

Table 2: Classification Report

4.2 Confusion matrix

The confusion matrix offers a comprehensive view of the model's performance in classifying instances into Negative, Neutral, and Positive categories. By analyzing the matrix, we can gain valuable insights into the model's effectiveness and areas that might require improvement.

For the Negative class, the model successfully classified 83% of the instances as Negative. However, there were some misclassifications: 8% of the actual Negative instances were classified as Neutral, and 10% were classified as Positive.

In the case of the Neutral class, the model accurately identified 77% of the instances. Yet, there were still some errors: 10% of the actual Neutral instances were classified as Negative, and 14% were classified as Positive.

The model performed exceptionally well regarding the Positive class, correctly classifying 92% of the instances. Only a small percentage of instances were misclassified: 4% of the actual Positive instances were classified as Negative, and another 4% were classified as Neutral.

Overall, the model demonstrates strong performance in classifying instances, especially in the Positive class, with an impressive classification rate of 92%. The model also effectively identifies the Negative class, achieving 83% accuracy. However, the model's performance in classifying the Neutral class could be improved, as its accuracy is 77%. The confusion matrix underscores the model's difficulty in differentiating between Negative and Neutral instances or between Neutral and Positive instances. This information is a valuable starting point for enhancing the model's performance.

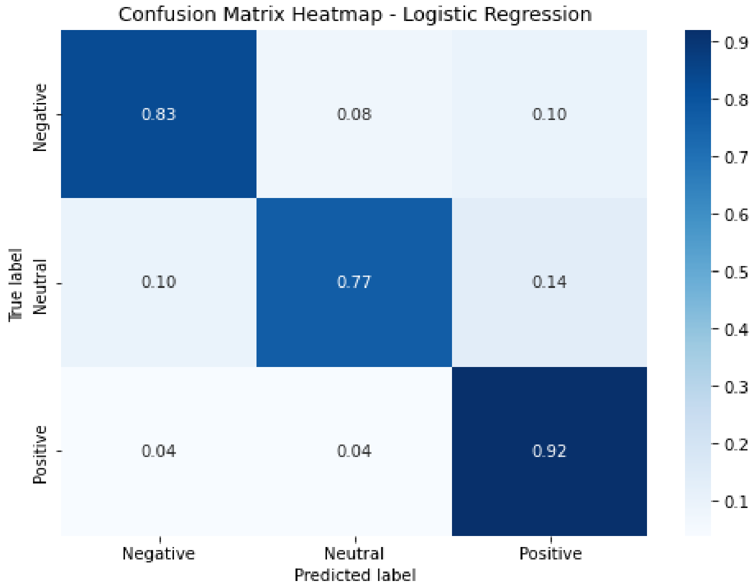


Figure 22: LR Confusion Matrix Heatmap

4.3 Receiver Operating Characteristic (ROC) Curve

Another means of visualizing the model's effectiveness is to utilize a ROC curve, a visual tool used to assess a classifier's performance by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR) at various decision thresholds. The Area Under the Curve (AUC) provides a single metric summarizing the classifier's performance, with higher AUC values (closer to 1) indicating better classification ability.

The Negative class has an AUC of 0.89, signifying a strong capability to distinguish between Negative instances and instances from other classes. This reveals the model's effectiveness in identifying Negative instances. The model's AUC is slightly lower at 0.86 in the Neutral class. Despite being a respectable score, the model might encounter challenges in accurately classifying Neutral instances compared to Negative and Positive instances. The model exhibits exceptional performance for the Positive class, with an AUC of 0.90. This high score demonstrates the model's strong ability to identify Positive instances.

The multi-class ROC curve results indicate that the model is proficient in classifying instances for all three classes, with the most notable performance for the Positive class. While the model effectively identifies Negative instances, its performance is somewhat weaker for the Neutral class. These results underscore the model's strengths and pinpoint areas where further improvements can be made to boost its classification capabilities.

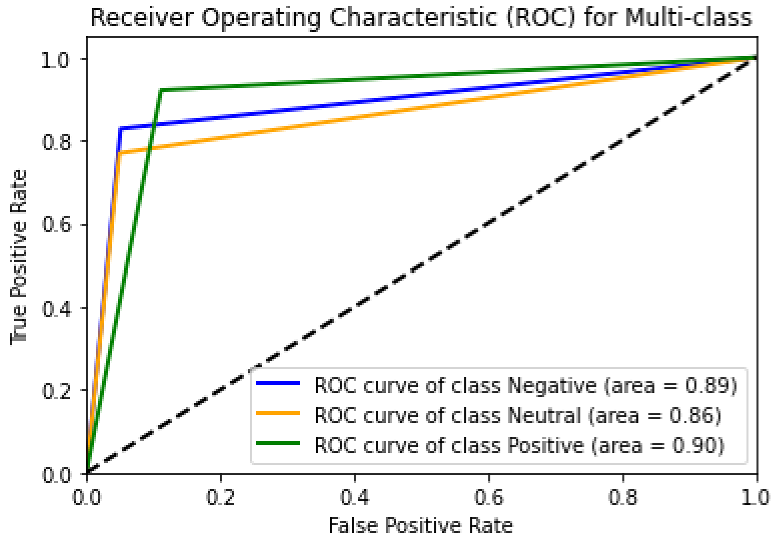


Figure 23: LR ROC Curve

4.4 Precision-Recall Curve

The Precision-Recall (PR) curve is a visual tool displaying a classifier's performance by assessing the balance between precision and recall at different decision thresholds. In multi-class situations, micro-averaging combines precision and recall values for each class, calculating average performance while accounting for class imbalance.

The micro-average PR curve for the model is 0.94, summarizing the classifier's performance in balancing precision and recall. A higher area (closer to 1) indicates better classification performance. The model demonstrates exceptional performance with a 0.94 area under the micro-average PR curve, maintaining high precision and recall across all classes. It effectively identifies instances while minimizing false positives and negatives, even with class imbalance or varying decision thresholds.

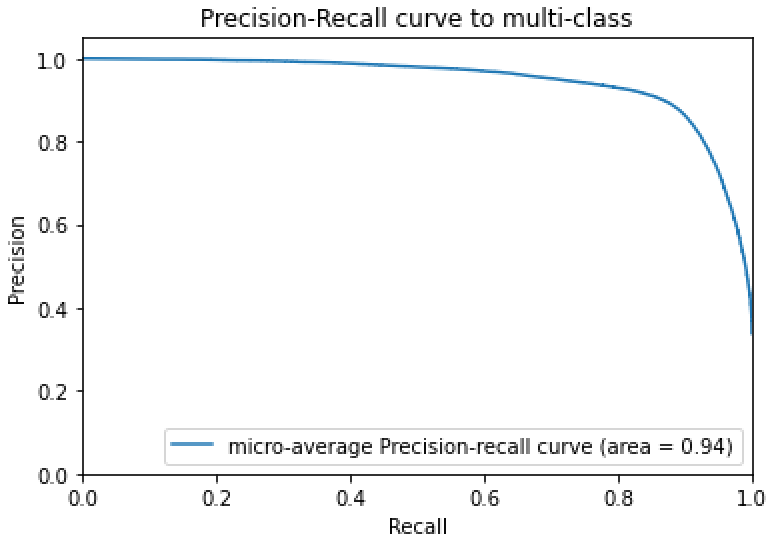


Figure 24: Logistic Regression PR Curve

5. Conclusion

This research project presents a comprehensive study of sentiment analysis applied to movie reviews from the Rotten Tomatoes website. The study demonstrates the value and utility of sentiment analysis in understanding the emotions and opinions of people toward movies.

The paper outlines a step-by-step process of data acquisition, preprocessing, exploration, model training, and model evaluation. The data preprocessing includes steps like lower-case conversion and text normalization, while the model training involves the use of machine learning algorithms like Naïve Bayes, Logistic Regression, and Support Vector Machines.

The study shows that the sentiment analysis model performs well in classifying the polarity of reviews as positive, negative, or neutral. Overall, the study establishes that sentiment analysis is a powerful tool that can be used by filmmakers, movie studios, and marketers to better understand their audience, inform their decision-making process, and improve future projects. The model can also serve as a benchmark for further research and improvements in the field of sentiment analysis on movie reviews.

References

Dataset

Leone, S. (2020). *Rotten Tomatoes movies and critic reviews dataset*. Retrieved from Kaggle: https://www.kaggle.com/datasets/stefanoleone992/rotten-tomatoes-movies-and-critic-reviews-dataset?resource=download&select=rotten_tomatoes_movies.csv

Python Libraries

Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830. https://scikit-learn.org/stable/

NLTK Project. (n.d.). NLTK 3.6.2 documentation. Retrieved from http://www.nltk.org/

Loria, S. (n.d.). TextBlob: Simplified Text Processing. Retrieved from https://textblob.readthedocs.io/en/dev/

Online Resources

Bird, S., Klein, E., & Loper, E. (2009). Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit. O'Reilly Media, Inc.

Scikit-Learn. (n.d.). Scikit-Learn User Guide. Retrieved from https://scikit-learn.org/stable/user_guide.html

Towards Data Science. (n.d.). Understanding ROC Curves. Retrieved from https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

Shah, P. (2020). Sentiment Analysis using TextBlob. Towards Data Science. Retrieved from https://towardsdatascience.com/my-absolute-go-to-for-sentiment-analysis-textblob-3ac3a11d524