A Comprehensive Report on Sentiment Analysis for Movie Reviews

This report delves into the application of sentiment analysis to movie reviews sourced from Rotten Tomatoes, a renowned platform for film critique and audience feedback. Using a dataset comprising 957,050 reviews, the project harnesses the power of natural language processing (NLP) and machine learning to classify sentiments as either positive or negative.

The analysis emphasizes key steps in data preparation, such as handling missing values, encoding sentiment labels, and preprocessing text to create a robust foundation for predictive modeling. Two distinct approaches were employed for sentiment classification:

1. VADER (Valence Aware Dictionary and sEntiment Reasoner): A lexicon-based method suited for capturing sentiment polarity in textual data.

2. Logistic Regression: A machine learning algorithm that uses TF-IDF vectorized features for sentiment prediction.

Model performance was rigorously evaluated using metrics such as accuracy, precision, recall, and F1 score. These metrics provided a comprehensive view of each model's strengths and limitations, offering insights into their real-world applicability.

The findings from this project uncover trends in public and critic sentiment, shedding light on patterns of positivity and negativity in movie reviews. By automating the sentiment analysis process, the study demonstrates potential applications for studios, critics, and audiences in understanding movie reception, guiding decision-making, and refining content strategies.

Introduction

Sentiment analysis (SA) has become an essential tool in natural language processing (NLP) for understanding and extracting insights from text data. This project leverages sentiment analysis to evaluate movie reviews from Rotten Tomatoes, aiming to predict the sentiment (positive or negative) of reviews based on textual content. Rotten Tomatoes, a popular platform for movie reviews, provides a rich dataset of critic reviews along with sentiment labels, making it an ideal source for this analysis.

This project focuses on applying sentiment analysis (SA) to movie reviews from Rotten Tomatoes, leveraging machine learning and NLP techniques to predict whether reviews are positive or negative. With 957,050 reviews in the dataset, including critic names, scores, content, and sentiment labels, this analysis provides valuable insights into public and critic perceptions.

The dataset underwent thorough preprocessing, including handling missing values and encoding sentiment labels for model training. By analyzing review text, the project demonstrates the scalability and accuracy of sentiment analysis in evaluating audience reactions, critic feedback, and patterns in movie reception. These insights can help studios, critics, and audiences better understand movie sentiments and public opinion.

Objective of the Project

The primary objective of this project is to **analyze and predict sentiment in movie reviews from Rotten Tomatoes**. Sentiment analysis is a critical tool in understanding the opinions and emotions expressed within a text. In the context of movie reviews, it allows us to categorize reviews into positive and negative sentiments, providing valuable insights into public perception and critical reception.

Rotten Tomatoes, a leading platform for movie reviews, offers a comprehensive collection of critic reviews that includes both textual content and associated sentiment labels. This dataset presents a unique opportunity to apply sentiment analysis techniques and machine learning algorithms to predict the sentiment of new, unseen reviews based on the language used.

The specific objectives of this project are:

1. **Data Exploration and Understanding:**
   To explore the Rotten Tomatoes dataset, analyze the features (such as review content, critic names, publication details), and understand the relationship between different attributes, including sentiment labels.
2. **Data Cleaning and Preprocessing:**
   To clean the dataset by handling missing values, removing irrelevant or incomplete data, and encoding categorical values, particularly the sentiment labels (e.g., 'Fresh' and 'Rotten'), into binary values for easier machine learning model implementation.
3. **Sentiment Prediction:**
   To build a machine learning model that predicts sentiment based on the content of the reviews.

This involves training the model on the review text and its corresponding sentiment label, then testing the model's accuracy in predicting sentiment on new reviews.

4. **Application of NLP Techniques:**
   To apply natural language processing (NLP) techniques to process the review text, including text normalization, tokenization, and vectorization, which prepares the text data for machine learning models.
5. **Evaluation and Insights:**
   To evaluate the performance of the sentiment analysis model using various evaluation metrics such as accuracy, precision, recall, and F1 score. Additionally, insights regarding the distribution of sentiments in the dataset, as well as trends in critic behavior, will be extracted from the analysis.

Ultimately, the project aims to create a reliable sentiment prediction model that can automatically classify movie reviews into positive or negative sentiment categories. This model could be useful for automating review classification, gaining insights into movie reception, and assisting studios, critics, and audiences in understanding how films are perceived based on reviews.

Description of the Dataset

The dataset used for this project is a comprehensive collection of movie reviews sourced from Rotten Tomatoes, featuring critic reviews, associated metadata, and sentiment labels. It provides an excellent foundation for sentiment analysis tasks due to its textual richness and structured attributes. With 957,050 rows and 8 columns, the dataset offers extensive data to uncover insights into reviewer behavior, sentiment trends, and scoring patterns.
Key features of the dataset include unique review links (rotten_tomatoes_link), critic details (critic_name), and a binary indicator of "Top Critic" status (top_critic). The dataset also captures publication details (publisher_name), review types (review_type, such as Fresh or Rotten), numerical review scores (review_score), publication dates (review_date), and the primary textual input for sentiment analysis (review_content).

Notably, the dataset contains 17,694 unique review links, 10,433 unique critics, and reviews from 2,198 unique publishers, with New York Times being the most frequent contributor. The reviews cover a broad timeline, with the oldest dating back to January 1, 2000, and offer valuable metadata for sentiment classification and predictive modeling.

While there are missing values in columns like critic_name (16,682 missing values) and review_content (61,944 missing values), the dataset’s overall scale and diversity remain significant strengths. This structured and detailed dataset serves as a robust resource for conducting sentiment analysis, revealing valuable insights into movie reviews and their associated sentiments.

Key Dataset Features
The dataset contains 957,050 rows and 8 columns, providing detailed information for analysis. Below is a summary of the dataset’s key features:

1. rotten_tomatoes_link
   Description: Unique links to Rotten Tomatoes pages for individual reviews.
   Statistics:
   - Total unique values: 17,694
   - Example: m/star_wars_the_rise_of_skywalker
2. critic_name
   Description: The name of the critic who authored the review.
   Statistics:
   - Total unique values: 10,433
   - Most frequent critic: Emanuel Levy (6,555 reviews)
   - Missing values: 16,682
3. top_critic
   Description: A binary indicator of whether the critic is a "Top Critic" on Rotten Tomatoes.
   Statistics:
   - Unique values: 4 (True, False, and possibly others)
   - Most frequent value: False (716,697 instances)
4. publisher_name
   Description: Name of the publication where the review appeared.
   Statistics:
   - Total unique values: 2,198
   - Most frequent publisher: New York Times (10,829 reviews)
5. review_type
   Description: Type of review provided (e.g., Fresh, Rotten).
   Statistics:
   - Total unique values: 45
   - Most frequent type: Fresh (591,912 reviews)
6. review_score
   Description: Numerical score assigned to the movie by the critic.
   Statistics:
   - Missing values: 250,895
   - Total unique values: 723
   - Most frequent score: 3/5 (76,390 instances)
7. review_date
   Description: Date when the review was published.
   Statistics:
   - Total unique dates: 8,031
   - Oldest review: January 1, 2000
   - Most frequent date: January 1, 2000 (41,155 reviews)
8. review_content
   Description: Text content of the review, serving as the main input for sentiment analysis.
   Statistics:
   - Total unique values: 728,606
   - Missing values: 61,944

Exploratory Data Analysis (EDA)

- Dataset Dimensions

  The dataset consists of 957,050 rows and 8 columns, providing a large-scale collection of movie reviews. Each row represents a unique review, and the columns capture various attributes, including critic information, review scores, and review content. The dataset's structure allows for comprehensive analysis and exploration of sentiment trends, critic behavior, and review patterns.

  Describing the overall dataset

```
df.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

Unique Values:

- The rotten_tomatoes_link column contains 17,694 unique links, each representing a specific movie review.
- There are 10,433 unique critics listed in the critic_name column, with Emanuel Levy being the most frequent critic, contributing 6,555 reviews.
- The review_type column shows 45 unique classifications, with Fresh being the most common sentiment, appearing in 591,912 reviews, indicating that the majority of reviews are positive.
- The review_score column has 723 unique values, with 3/5 being the most frequent rating, appearing 76,390 times.
- The review_date spans 8,031 unique dates, with 2000-01-01 being the most frequent review date, contributing 41,155 reviews.

Most Reviewed Movie:

- Star Wars: The Rise of Skywalker holds the record for the most reviewed movie in the dataset, with a total of 916 reviews.

Top Publication:

- The New York Times is the most frequent publisher, contributing a total of 10,829 reviews to the dataset.

Imbalance in Data:

- The top_critic column reveals an imbalance in the data, with 716,697 entries marked as False (non-top critics), indicating that most of the reviews come from non-top critics.
- The review_type column also reflects an imbalance, as it is dominated by Fresh reviews, highlighting that the dataset has a higher number of positive sentiment reviews compared to

negative ones. This imbalance could impact sentiment analysis models and may require techniques like resampling to address during the modeling process.

Displaying first five rows of the dataset:

First 5 rows of the dataset:

	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...

Insights from EDA

1. Textual Data:
The dataset's `review_content` column offers a rich source of text for sentiment analysis but has missing values that must be addressed.
2. Numeric Conversion:
The `review_score` column, though stored as `object`, contains numeric values (e.g., `3/5`) that require conversion for analysis.
3. Missing Data:
Key columns like `review_score` and `review_content` have missing values, necessitating imputation or removal strategies.
4. Bias in Data:
Certain critics and publishers are overrepresented, which could influence sentiment analysis results.
5. Potential Applications:
The dataset is suitable for sentiment analysis, enabling comparisons between lexicon-based methods (e.g., VADER) and machine learning classifiers (e.g., Logistic Regression).

This exploratory analysis provides a solid foundation for preprocessing and further analysis, ensuring data readiness for sentiment modeling and classification.

Data Cleaning and Preprocessing

Checking for Missing Values

The dataset was examined for missing values in the most important columns for sentiment analysis:

Observations:

1. Complete Columns: Columns like `rotten_tomatoes_link`, `top_critic`, and `publisher_name` are fully populated with no missing values.

2. Missing Values:

`critic_name` has 16,682 missing entries.

`review_score` has 250,895 missing entries.

`review_content` has 61,944 missing entries.

3. Dropping Rows with Missing Values

To maintain data quality and accuracy, rows containing missing values in the review_content or review_type columns were discarded. This resulted in a reduction of the dataset from 957,050 rows to 872,671 rows, ensuring that only complete and valid records remained for analysis.

```
Data types of columns after dropping missing values:
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
```

4. **Encoding the review_type Column:**

The review_type column, which contains categorical values such as 'Fresh' and 'Rotten', was encoded into binary values for easier analysis:

- 'Fresh' was encoded as 1 (indicating positive sentiment).
- 'Rotten' was encoded as 0 (indicating negative sentiment).

This encoding step transformed the categorical sentiment labels into numerical values, making the dataset suitable for machine learning models and sentiment analysis tasks.

5. **Validation of Encoding**

After encoding the review_type column, the dataset was rechecked to ensure that no missing values were present in this column. The validation confirmed that all sentiment labels had been correctly encoded, and no missing values remained.

6. **Preview of the Cleaned Data**

The cleaned dataset, now consisting of 872,671 rows, was previewed to confirm the changes. The review_content and review_type columns were inspected, and it was confirmed that the missing values had been removed and the encoding was correct. The cleaned dataset is now ready for further analysis, such as sentiment classification or machine learning model training.

Cleaned Dataset Preview: The cleaned dataset contains the following columns:

- review_content: The textual content of the movie review.
- review_type: The sentiment label (1 for 'Fresh', 0 for 'Rotten').

```
..
    Cleaned Dataset Info:
    <class 'pandas.core.frame.DataFrame'>
    Index: 1064211 entries, 0 to 1130016
    Data columns (total 8 columns):
     #   Column                Non-Null Count    Dtype
    ---  ------                --------------    -----
     0   rotten_tomatoes_link  1064211 non-null  object
     1   critic_name           1054198 non-null  object
     2   top_critic            1064211 non-null  bool
     3   publisher_name        1064211 non-null  object
     4   review_type           1064211 non-null  int64
     5   review_score          758709 non-null   object
     6   review_date           1064211 non-null  object
     7   review_content        1064211 non-null  object
    dtypes: bool(1), int64(1), object(6)
    memory usage: 66.0+ MB
    None
```

With the missing data handled and categorical values encoded, this dataset is now in an optimal

Sentiment Analysis Overview

- Explanation of Sentiment Analysis
- Techniques Used: Lexicon-Based (VADER) and Machine Learning

Lexicon-Based Sentiment Analysis

Sentiment Analysis with VADER

1. Initializing VADER Sentiment Analyzer

VADER (Valence Aware Dictionary and sEntiment Reasoner) is a pre-trained sentiment analysis tool, optimized for social media text such as reviews. The SentimentIntensityAnalyzer from the vaderSentiment library was initialized to analyze the sentiment of review content.

2. Applying VADER Sentiment Analysis: The VADER sentiment analysis was applied to the review_content column to compute the compound sentiment score for each review. The compound score, which ranges from -1 to 1, was used to classify sentiment:

- Scores close to 1 indicate positive sentiment.
- Scores close to -1 indicate negative sentiment.
- Scores around 0 indicate neutral sentiment.

The computed compound scores were stored in a new column, vader_score.

3. Converting VADER Scores to Binary Sentiments: The compound scores were converted into binary sentiment labels:

- Sentiment scores greater than or equal to 0 were classified as positive sentiment (1).
- Scores less than 0 were classified as negative sentiment (0).

The result of this conversion was stored in the vader_pred column, representing the binary sentiment predictions for each review.

4. Handling Missing Data: Rows with missing values in the review_type or vader_pred columns were removed to ensure the dataset's integrity for analysis.

5. Ensuring Integer Data Types: The review_type and vader_pred columns were converted to integer types to ensure compatibility with analysis and evaluation metrics.

6. Classification Report: A classification report was generated using scikit-learn's classification_report, comparing the true sentiments (review_type) with the predicted sentiments (vader_pred). The report includes precision, recall, F1-score, and support for both positive and negative sentiment classes.

Classification Report:

VADER Sentiment Analysis Report:

	precision	recall	f1-score	support
0	0.54	0.34	0.42	70961
1	0.67	0.82	0.74	115136
accuracy			0.64	186097
macro avg	0.60	0.58	0.58	186097
weighted avg	0.62	0.64	0.62	186097

- **Precision** indicates the accuracy of the positive/negative sentiment predictions.
- **Recall** shows how many of the actual sentiments were correctly identified.
- **F1-score** provides a balance between precision and recall.

Conclusion:

- The model performs better on positive reviews (1) with higher recall and F1-score.
- The model's performance on negative reviews (0) is lower, indicated by a lower recall and F1-score.

7. Model Accuracy: The overall accuracy of the VADER sentiment analysis model was approximately **64%**, calculated using the accuracy_score function from scikit-learn.

8. Sample of Transformed Data: A sample of the transformed dataset was displayed, showing the original review_content, the true sentiment (review_type), the computed VADER sentiment score (vader_score), and the predicted sentiment (vader_pred).

Sample of Transformed Data:

review_content	review_type	vader_score	vader_pred
A fantasy adventure that fuses Greek mythology...	1	0.7579	1
It's more a list of ingredients than a movie-m...	0	0.4939	1
Harry Potter knockoffs don't come more transparent...	0	0.0000	1
The best thing you can say about Chris Columbus'...	1	-0.2500	0
This cast is simply too generic. None of the...	0	0.0000	1

Machine Learning Approach

Feature Extraction for Logistic Regression (TF-IDF)

1. Encoding the Target Variable: Before training the machine learning models, the target variable, review_type, was encoded into numeric format using LabelEncoder. This transformation is necessary because machine learning algorithms work with numeric data, and the target variable (which represents sentiment) is categorical.
Since the task involves classifying reviews into two categories—positive sentiment and negative sentiment—the target variable was encoded as:

- 0: Negative sentiment
- 1: Positive sentiment

This encoding allows the model to process sentiment as numeric labels, facilitating the learning process.

Classes in Target Variable: After encoding, the classes in the target variable were displayed to confirm the encoding was performed correctly. The output confirmed that the transformation was accurate, with two classes: 0 (negative sentiment) and 1 (positive sentiment).

2. Initializing TF-IDF Vectorizer

The TfidfVectorizer was initialized to extract features from the review content. TF-IDF (Term Frequency-Inverse Document Frequency) is a statistical measure used to evaluate how important a word is to a document in a corpus. The settings used for the vectorizer were as follows:

- max_features=5000: This limits the vocabulary to the 5000 most frequent words across the reviews, ensuring the model uses only the most relevant terms.
- stop_words='english': Commonly used English stopwords (e.g., "the," "is," "in") were removed, as they do not provide significant information for sentiment analysis.
- ngram_range=(1, 2): This setting allows the extraction of both unigrams (single words) and bigrams (two consecutive words) as features. This approach helps capture more contextual meaning, especially for phrases that convey sentiment (e.g., "not good" or "very happy").
- max_df=0.7: Words that appear in more than 70% of the documents were excluded, as they tend to be too common and may not carry useful information for classification.
- min_df=5: Words that appear in fewer than 5 documents were excluded to eliminate rare words that are unlikely to contribute to the model's predictive power.

3. Transforming the Review Content Using TF-IDF

Once the TfidfVectorizer was initialized, it was applied to the review_content column to create a feature matrix. This matrix represents the TF-IDF values for each word or bigram in the reviews. Each entry in the matrix corresponds to the importance of a particular word or phrase in a specific review, based on its frequency and the frequency of that word across all reviews.

Number of Features Extracted: A total of 5000 unique features (words and n-grams) were extracted from the review content. These features form the basis for the model's learning process.

4. Splitting the Data into Training and Testing Sets

To evaluate the model's performance, the dataset was split into training and testing sets. A common approach is to reserve a portion of the data for training the model and another portion for testing its performance on unseen data. The data split was done using train_test_split, with a stratified distribution to ensure that the proportion of positive and negative reviews remained consistent across both training and testing sets.

- Training Data (80%): 80% of the dataset was used to train the model, consisting of 148,877 reviews.
- Testing Data (20%): The remaining 20% of the data was reserved for testing, comprising 37,220 reviews.

Dataset Statistics:

- Feature Matrix Shape: The TF-IDF feature matrix has a shape of (186,097, 5000), meaning there are 186,097 reviews and 5000 unique features (words or n-grams).
- Training Data Shape: The training data contains 148,877 reviews and 5000 features.
- Testing Data Shape: The testing data contains 37,220 reviews and 5000 features.

Class Distribution:

The distribution of reviews by sentiment across the training and testing sets was as follows:

- Training Data: The training set contains 56,769 negative reviews (labeled as 0) and 92,108 positive reviews (labeled as 1).
- Testing Data: The testing set contains 14,192 negative reviews (labeled as 0) and 23,028 positive reviews (labeled as 1).

This indicates that the dataset is slightly imbalanced, with a larger number of positive reviews than negative reviews, both in the training and testing sets.

Conclusion of Data Preprocessing

At this stage, the data preprocessing steps have been completed successfully:

- The target variable (review_type) has been encoded into numeric labels for machine learning.
- The review content has been transformed into a TF-IDF feature matrix that represents the important terms (unigrams and bigrams) within the reviews.
- The dataset has been split into training and testing sets, ensuring a representative distribution of sentiment labels across both sets.

This data is now ready for the next steps, which involve training machine learning models and evaluating their performance on sentiment classification tasks.

Logistic Regression Model Training and Evaluation

1. Initializing the Logistic Regression Model

The Logistic Regression model was initialized with specific parameters to optimize performance for sentiment analysis on the dataset. Below are the detailed configurations used:

- max_iter=200: The maximum number of iterations was set to 200, which ensured that the optimization algorithm had enough cycles to converge effectively, especially given the large size and complexity of the dataset. Without sufficient iterations, the model might have failed to converge or provided suboptimal results.
- solver='saga': The 'saga' solver, known for its efficiency with large, sparse datasets, was selected. This solver is particularly useful for scenarios involving high-dimensional data, ensuring faster convergence compared to traditional solvers.
- penalty='l2': L2 regularization was applied to the model to mitigate overfitting. This regularization method penalizes large coefficients, effectively balancing model complexity and performance on unseen data.
- n_jobs=-1:The n_jobs parameter was set to -1 to utilize all available CPU cores. This parallelization strategy significantly reduced the training time by distributing computations across multiple processors.

2. Training the Logistic Regression Model: The training process involved fitting the Logistic Regression model to the training dataset, which comprised 80% of the total data. During this phase, the model learned to map input features (movie reviews) to their corresponding sentiment labels (positive or

negative) by minimizing the prediction error through iterative adjustments of its internal parameters. The fit method was used to execute this training process.

3. Making Predictions: Once the model was trained, predictions were generated using the test dataset, which accounted for the remaining 20% of the data. The predict method was employed to assign sentiment labels (positive or negative) to each review in the test set based on the patterns learned during training.

4. Evaluating Model Performance: To assess the effectiveness of the Logistic Regression model, a classification report was generated. This report summarized key metrics, including precision, recall, F1-score, and overall accuracy, for both classes (positive and negative sentiments). The detailed results are presented below:

Logistic Regression Classification Report:

	precision	recall	f1-score	support
0	0.71	0.56	0.62	14192
1	0.76	0.86	0.80	23028

Interpretation of Evaluation Metrics

Precision

- **Class 0 (Rotten Reviews):**
  Precision of 0.71 indicates that 71% of reviews predicted as "Rotten" were actually "Rotten." This suggests that while the model is relatively reliable in identifying negative reviews, there is room for improvement in reducing false positives.
- **Class 1 (Fresh Reviews):**
  Precision of 0.76 means that 76% of reviews predicted as "Fresh" were correctly labeled as such. This higher precision for positive reviews reflects the model's stronger ability to identify positive sentiment.

Recall

- **Class 0 (Rotten Reviews):**
  Recall of 0.56 indicates that the model correctly identified 56% of the actual "Rotten" reviews. This lower recall suggests that the model struggles to detect all negative reviews, leading to a higher number of false negatives.

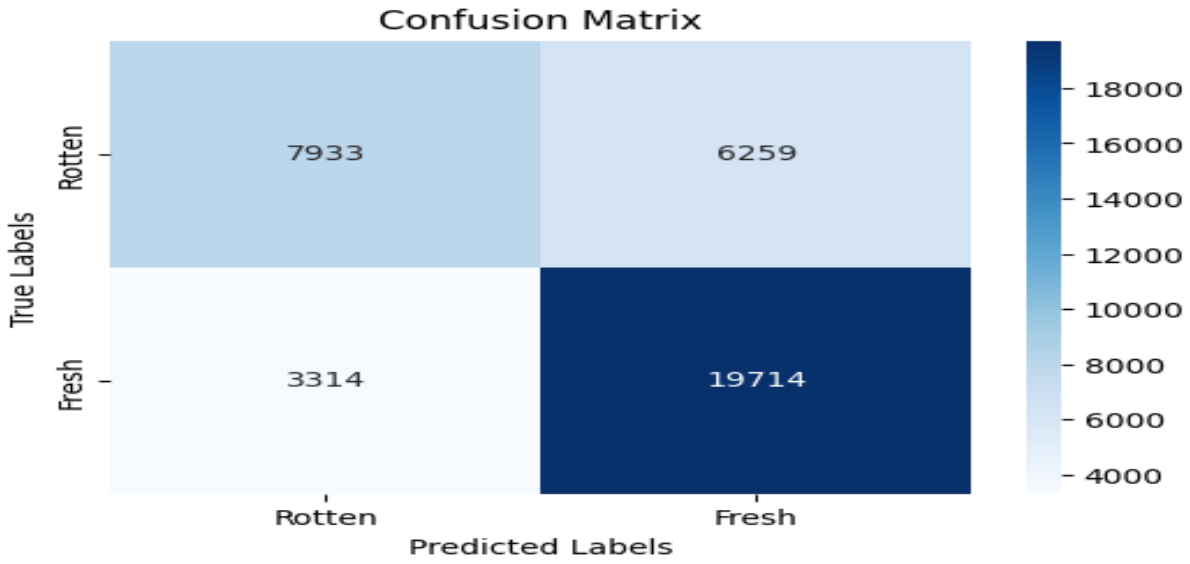
- **Class 1 (Fresh Reviews):**
  Recall of 0.86 shows that 86% of actual "Fresh" reviews were correctly classified. This highlights the model's strong performance in capturing positive sentiment.

F1-Score

- **Class 0 (Rotten Reviews):**
  An F1-score of 0.62 represents a balance between precision and recall but indicates that the model has moderate effectiveness in identifying negative reviews accurately.
- **Class 1 (Fresh Reviews):**
  An F1-score of 0.80 demonstrates the model's strong overall performance in predicting positive reviews, effectively balancing both precision and recall.

Confusion Matrix Interpretation

The confusion matrix provides a detailed breakdown of the model's predictions:



- **True Positives (Fresh predicted as Fresh): 19,714**
  - The number of positive reviews correctly classified as "Fresh."
- **True Negatives (Rotten predicted as Rotten): 7,933**
  - The number of negative reviews correctly classified as "Rotten."
- **False Positives (Rotten predicted as Fresh): 6,259**
  - The number of negative reviews incorrectly classified as "Fresh."
- **False Negatives (Fresh predicted as Rotten): 3,314**
  - The number of positive reviews incorrectly classified as "Rotten."

Conclusion: The Logistic Regression model achieved an overall accuracy of 74%, reflecting its effectiveness in predicting sentiment labels for the test dataset. While the model performed exceptionally well in identifying positive reviews, as evidenced by its high precision, recall, and F1-score for Class 1

(Fresh reviews), it exhibited weaknesses in detecting negative reviews, with lower recall and F1-score for Class 0 (Rotten reviews).
Despite these limitations, the model demonstrates potential for practical applications in sentiment analysis, particularly for tasks requiring reliable identification of positive sentiment. Further refinements, such as adjusting hyperparameters, balancing the dataset, or exploring alternative algorithms, may enhance its performance for negative reviews.

Performance Comparison: VADER vs Logistic Regression

When comparing sentiment analysis models, it is essential to evaluate their performance across various metrics to determine their suitability for specific tasks. This report presents a performance comparison between two models, **VADER** and **Logistic Regression**, for review classification. Both models were assessed based on **accuracy**, **precision**, **recall**, and **F1-score**, providing insights into their effectiveness and reliability. The goal of this comparison is to determine which model is better suited for accurately classifying reviews as "Fresh" or "Rotten," highlighting each model's strengths and weaknesses.

1. Model Accuracy

VADER Accuracy: 0.64

- The VADER model achieved an accuracy of **64%**, correctly predicting 64% of the reviews in the dataset.

Logistic Regression Accuracy: 0.74

- Logistic Regression achieved a higher accuracy of **74%**, correctly classifying 74% of the reviews in the test dataset.
- This demonstrates that Logistic Regression outperforms VADER in terms of overall accuracy.

2. Performance Metrics Comparison

- **VADER**: Achieved a precision of **0.67** for classifying positive reviews (Fresh).
- **Logistic Regression**: Recorded a higher precision of **0.76**, demonstrating a better ability to identify true positives.

Recall:

- **VADER**: Attained a recall of **0.82**, successfully capturing 82% of the actual positive reviews.
- **Logistic Regression**: Outperformed VADER with a recall of **0.86**, identifying a larger proportion of actual Fresh reviews.

F1-Score:

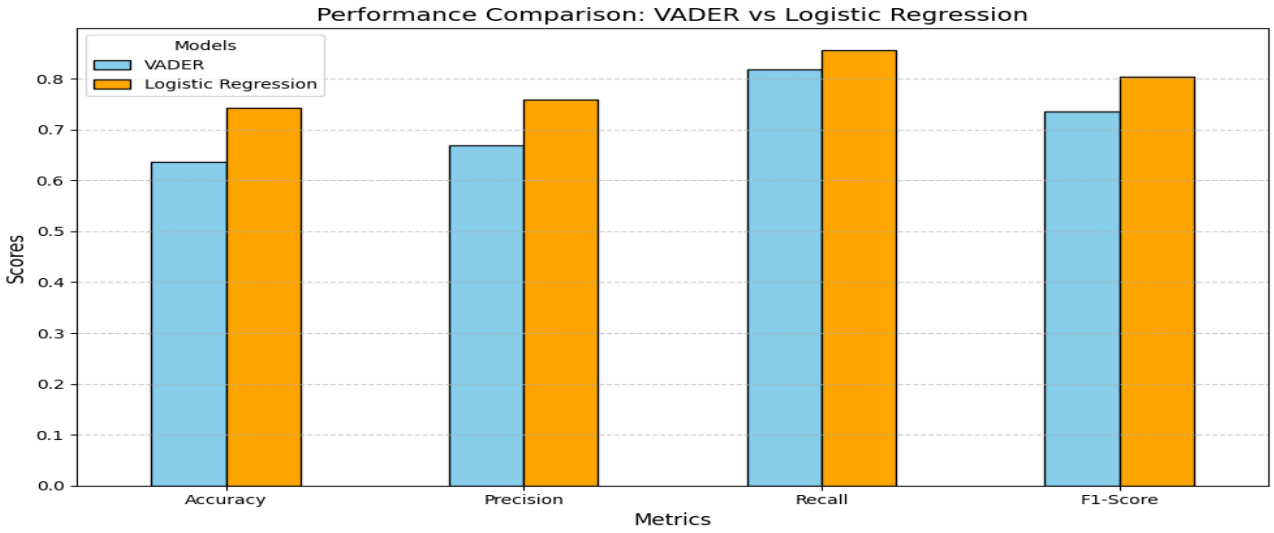
- **VADER**: Achieved an F1-score of **0.74**, indicating a balance of precision and recall for Fresh reviews.

- **Logistic Regression**: Scored **0.80**, reflecting stronger overall performance with an improved balance between precision and recall.

Model	Accuracy	Precision	Recall	F1-Score
VADER	0.64	0.67	0.82	0.74
Logistic Regression	0.74	0.76	0.86	0.80

3. Visualization of Results

A bar chart provides a clear comparison of the performance metrics for VADER and Logistic Regression.



Insights from the Chart:

- Logistic Regression outperforms VADER across all metrics:
  - **Accuracy**: Higher for Logistic Regression.
  - **Precision**: Improved precision for Logistic Regression.
  - **Recall**: Logistic Regression significantly exceeds VADER.
  - **F1-Score**: Logistic Regression demonstrates superior performance.

The chart visually reinforces the superior performance of Logistic Regression, making it the preferred model for review classification.

4. Conclusion

The performance comparison reveals the following insights:

1. Logistic Regression outperforms VADER across all metrics: **accuracy**, **precision**, **recall**, and **F1-score**.
2. Logistic Regression excels at identifying both "Fresh" and "Rotten" reviews more accurately.

3. VADER shows a slightly higher recall for Fresh reviews but struggles with precision, leading to lower overall performance.
4. VADER serves as a **baseline sentiment analysis model**, particularly for smaller datasets or scenarios prioritizing computational speed.
5. Logistic Regression offers a better balance of precision, recall, and accuracy, making it the **recommended model** for tasks where reliability and accuracy are essential.

5. Final Thoughts

- Logistic Regression is a more robust and reliable choice for review classification due to its superior performance across all metrics.
- VADER, with its higher recall, may still be a viable option in scenarios prioritizing speed or when working with limited data.
- For applications requiring high accuracy and balanced performance, Logistic Regression stands out as the stronger candidate.