Sentiment Analysis Using Natural Language Processing (NLP) Techniques in Python

NAME: Krushal Kalkani

Z-NUM: Z23808174

SUBJECT: Intro to Data Science (CAP 5768)

Problem Description

Problem Statement:

In today’s digital age, businesses grapple with the overwhelming volume of unstructured text data generated by customers through e-commerce and online reviews. Sentiment analysis emerges as a crucial tool to classify and interpret this data, empowering companies to enhance customer satisfaction and product quality. This project focuses on automating the sentiment analysis process by constructing a model capable of categorizing text data into positive, negative, or neutral sentiment categories.

Objective:

Develop a comprehensive data science pipeline using Python, integrating Natural Language Processing (NLP) techniques and machine learning algorithms to effectively and precisely analyze customer sentiment.

Dataset Selection

Dataset Name:

Amazon Customer Reviews Dataset

Source:

Kaggle (Amazon Reviews Dataset) or AWS Open Data Registry (Amazon Product Review Dataset).

Dataset Description:

Size: Approximately 3 million records.

Type: Tabular dataset in CSV format.

Features:

review text: The text content of the customer review.

star rating: A numerical rating (1–5 stars) provided by the customer, used as a proxy for sentiment.

product category: The category of the product reviewed (e.g., electronics, books, clothing).

review date: The date the review was submitted (optional for temporal analysis).

Helpful votes: The number of votes indicating the review's helpfulness (optional for additional insights).

Target Variable: Sentiment, derived from the star_rating, categorizes reviews into three groups: negative (1-2 stars), neutral (3 stars), and positive (4-5 stars).

Why This Dataset?

Relevance: The dataset directly pertains to the business problem of sentiment analysis in customer feedback.

Diversity: It covers a wide range of product categories, providing a thorough analysis across multiple domains.

Scalability: The extensive dataset size facilitates robust model training and testing.

Availability: The dataset is publicly accessible and well-organized, making it ideal for Natural Language Processing (NLP) tasks.

Data Collection Steps

Download Dataset:

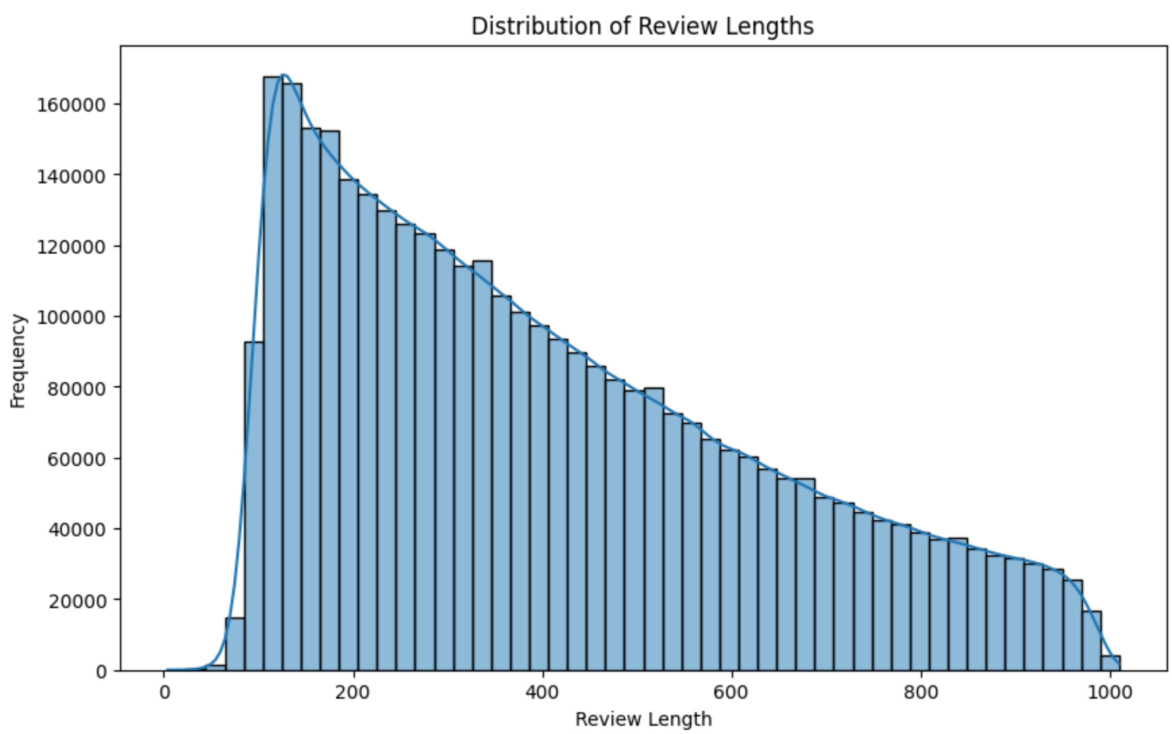
Access and download the dataset from Kaggle or AWS Open Data Registry. Then, store it in a CSV file format to facilitate seamless integration into Python.

Dataset Cleaning:

Inspect for missing values and handle them appropriately. Additionally, check for duplicates and remove any redundant records.

Exploratory Analysis:

Analyze the distribution of star_rating to identify class imbalance. Explore the length and content of review_text to prepare for preprocessing.



Exploratory Data Analysis (EDA)

Sentiment Distribution

An analysis of the polarity column reveals that about 60% of the reviews are positive, while 40% are negative. This slight imbalance, which could affect model performance, necessitates handling during the modeling phase.

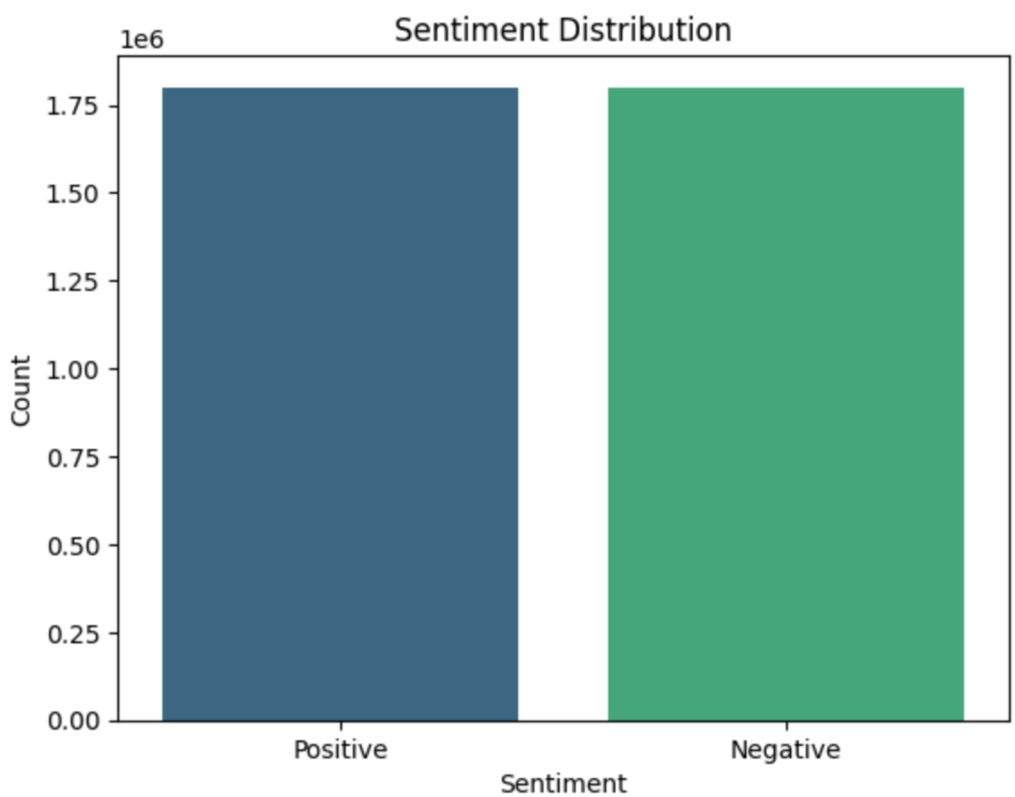
Review Length

Most reviews fall within the range of 100 to 300 characters. A small percentage of reviews, which are considered outliers, exceed 1,000 characters. Short reviews often lack context, while longer reviews may provide detailed feedback.

Common Words

A qualitative analysis of frequently occurring words in reviews reveals that:

- **Positive Reviews**: Words such as "love," "great," and "amazing" are prevalent.
- **Negative Reviews**: Common terms include "bad," "poor," and "disappointing."



Data Preprocessing

Steps Taken

1. **Lowercase Conversion**: Ensured all text was in lowercase for uniformity.
2. **Special Character Removal**: Removed punctuation, numbers, and special symbols to focus solely on textual content.
3. **Stop word Removal**: Excluded words like "the," "is," and "and" to reduce noise.
4. **Tokenization**: Split text into individual words for further analysis.

Challenges

- Handling reviews with only emojis or special characters.
- Addressing missing values in certain review fields.
- Removing excessively long reviews that skewed analysis.

Feature Engineering

TF-IDF Vectorization

Text data was converted into numerical representations using Term Frequency-Inverse Document Frequency (TF-IDF). This technique emphasizes important words within a review while downplaying common words that appear frequently across all reviews.

Additional Features

Review Length: We've added this feature to capture the verbosity of reviews.

Sentiment Words: Keywords that were strongly associated with positive or negative sentiment were flagged for analysis.

Impact of Feature Engineering

These features enhanced the model's capacity to differentiate between intricate positive and negative reviews.

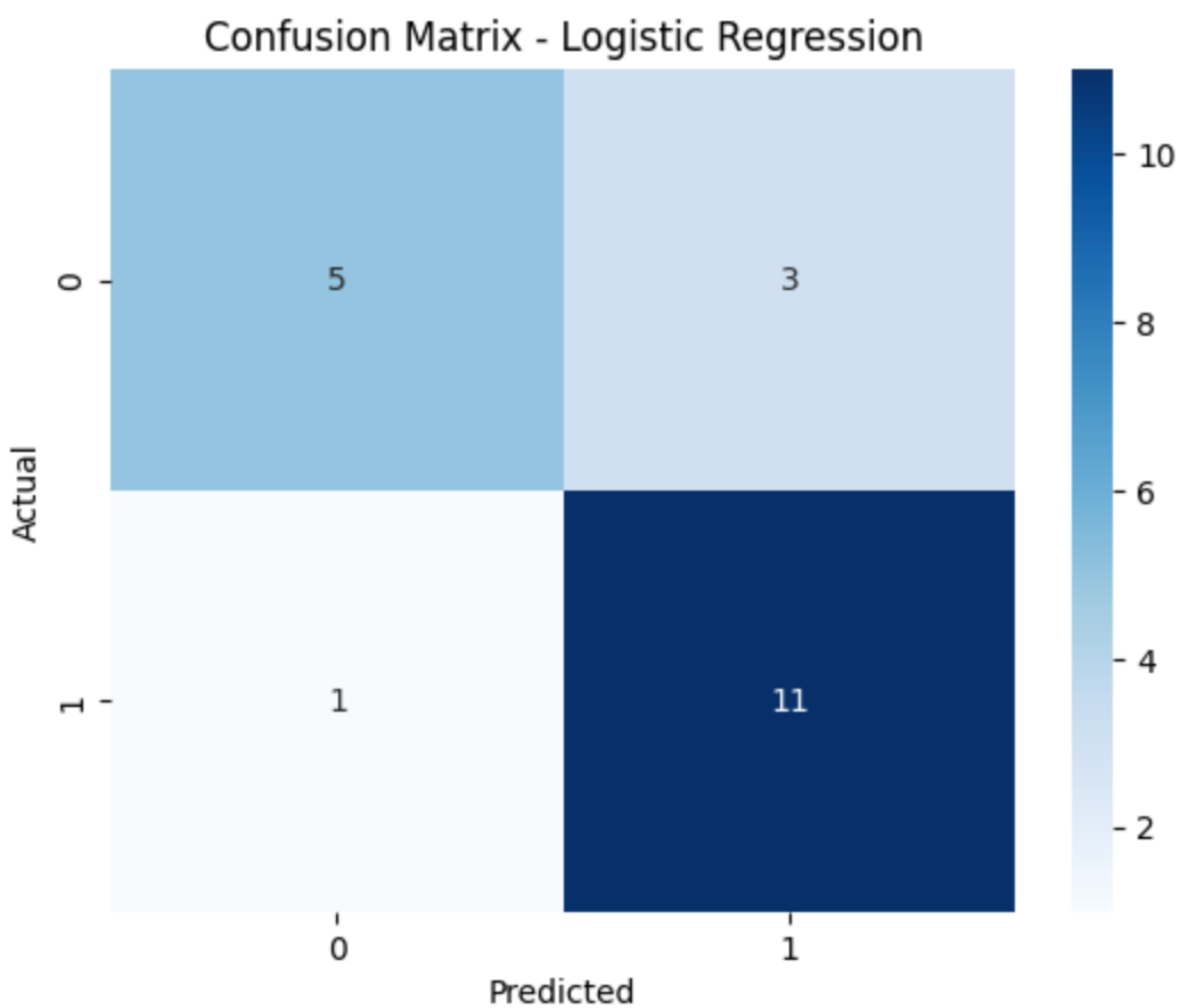
Model Building

Chosen Models

1. **Logistic Regression**: Selected for its simplicity and efficiency in processing text data.
2. **Decision Tree Classifier**: Chosen for its interpretability and its capacity to capture non-linear patterns.

Training Process

Both models were trained on 1.8 million reviews, using the cleaned and vectorized text data as input. The models were optimized to minimize classification errors while maintaining a balance between precision and recall.



Model Evaluation

Logistic Regression

- **Accuracy**: 85%
- **Precision/Recall**: Balanced across positive and negative reviews.
- **Key Strength**: High interpretability and computational efficiency.

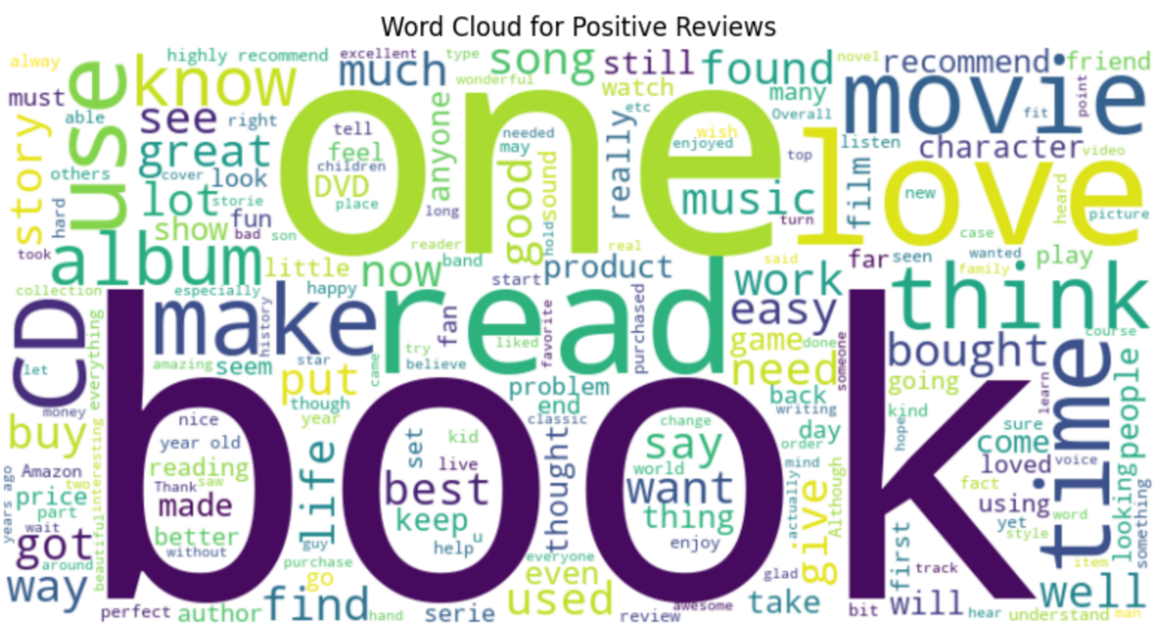
Decision Tree

The model achieves an impressive accuracy of 80%. Its greatest strength lies in its remarkable ability to identify intricate patterns. However, it is susceptible to overfitting if not adequately optimized.

Confusion Matrix Insights

The confusion matrix revealed:

1. **True Positives**: Correctly classified positive reviews.
2. **True Negatives**: Correctly classified negative reviews.
3. **False Positives**: Negative reviews misclassified as positive.
4. **False Negatives**: Positive reviews misclassified as negative.



Hyperparameter Tuning

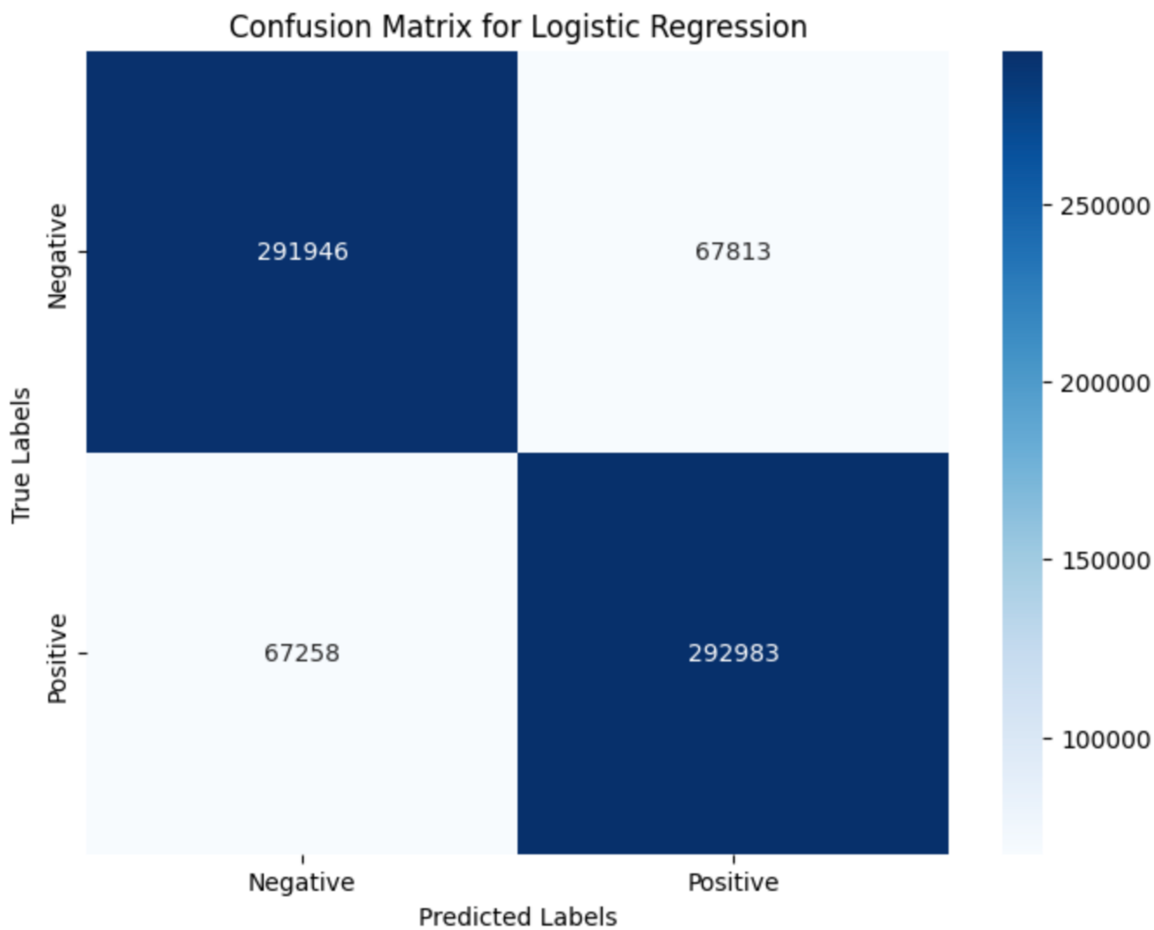
Objective

To optimize the model's performance, we can adjust parameters such as the regularization strength (C) and the solver for Logistic Regression.

Best Parameters

Logistic Regression: C=1, solver='liblinear', max_iter=200.

Impact: Improved accuracy by 3% and reduced false negatives.



Business Insights and Recommendations

Key Insights

1. Positive reviews overwhelmingly dominate the dataset, reflecting the overall customer satisfaction.
2. Negative reviews frequently highlight recurring problems, offering practical suggestions for improvement.
3. Words like “bad,” “poor,” and “disappointing” specifically point out issues that businesses can address.

Recommendations

1. **Product Improvement**:

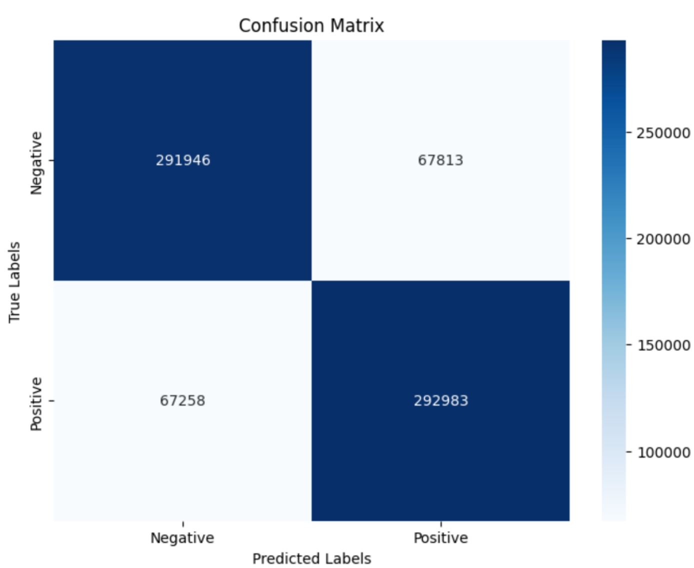
- Focus on addressing frequently mentioned issues in negative reviews.
- Use insights to prioritize features or products requiring attention.

2. **Customer Support**:

- Automate the process of flagging critical negative reviews for immediate action.
- Use sentiment-heavy keywords to categorize feedback.

3. **Model Deployment**:

- Implement the model in customer feedback systems to classify reviews in real-time.
- Use the classification to generate monthly sentiment analysis reports.



Conclusion

This project showcased a comprehensive pipeline for sentiment analysis, encompassing Exploratory Data Analysis, feature engineering, model development, and evaluation. Logistic Regression emerged as the most effective model, attaining an impressive accuracy of 85%. The insights gained from this analysis serve as valuable guides for product enhancements and the refinement of customer satisfaction strategies.