Sentiment Analysis of Customer Reviews

Hamza Salah – DSC 550 Data Mining

Introduction

This term project focuses on tackling a key issue through detailed analysis and model development, offering valuable insights and practical solutions for an Ecommerce platform like Amazon.

One of the questions raised during the course of the project involved the practical application of machine learning in classifying appliance reviews effectively. To address this question, the following approach was proposed:

Implementation of a machine learning model involves several key steps to ensure accurate classification of appliance reviews, encompassing both positive and negative feedback. Firstly, data preprocessing is essential; this includes cleaning the data, removing noise, and converting textual data into numerical representations using techniques such as TF-IDF (Term Frequency-Inverse Document Frequency).

Next, selecting and training an appropriate machine learning algorithm is critical. Algorithms such as Logistic Regression, designed for natural language processing tasks have shown promise in text classification. These models are trained on labeled datasets, learning to distinguish between positive and negative reviews based on features extracted during preprocessing.

Additionally, evaluating model performance using metrics such as accuracy, precision, recall, and F1-score provides insights into the model's effectiveness in classifying reviews accurately.

Introducing the Problem

How can we implement a machine learning model to accurately classify appliance reviews, ensuring strong performance for both positive and negative feedback?

For this project, we will be analyzing a large-scale Amazon Reviews dataset from 2023, which contains 48.19 million items and 571.54 million reviews. Our primary focus will be on the Appliances category within this dataset.

The Appliances category includes:

- 1.8 million unique users who have left reviews.

- 94.3K unique items in the Appliances category.
- 2.1 million customer reviews.
- 92.8 million ratings given for these products.
- 95.3 million total feedback entries.

The goal is to identify factors driving customer satisfaction or dissatisfaction across various product categories. Using NLP techniques like TextBlob and VADER, sentiment polarity scores will be extracted from reviews and categorized into positive (satisfied) or negative (dissatisfied).

A classification model, such as Logistic Regression, will be trained to predict sentiment based on review text. The target of the model is the binary sentiment (positive or negative) derived from polarity scores as well as the rating scale.

Justify Why It Is Important/Useful to Solve This Problem

Solving this problem is crucial for several reasons. First, it will lead to improved customer satisfaction insights, allowing manufacturers and retailers to adjust their product strategies. Second, addressing this issue will result in better-targeted marketing and product development. Lastly, finding a solution to this problem will pave the way for future advancements in sentiment analysis and NLP applications within e-commerce.

Pitch to Stakeholders

To gain buy-in from stakeholders, I emphasized the potential benefits of solving this problem. By accurately classifying sentiment in appliance reviews, we can expect to see improved customer experience through targeted product improvements and more effective marketing strategies. Furthermore, the solution will position our organization as a leader in utilizing advanced data science techniques to drive innovation and growth.

Data Acquisition

The data for this project was obtained from:

- Amazon Reviews 2023 dataset

This source offers comprehensive details, including product ratings, review texts, and timestamps, providing a rich foundation for analysis. Its extensive nature and reliability make it an ideal resource for uncovering trends and patterns that can drive improvements in both customer satisfaction and business performance.

Organized and Detailed Summary of Milestones 1-3

Milestone 1: Problem Definition and Research

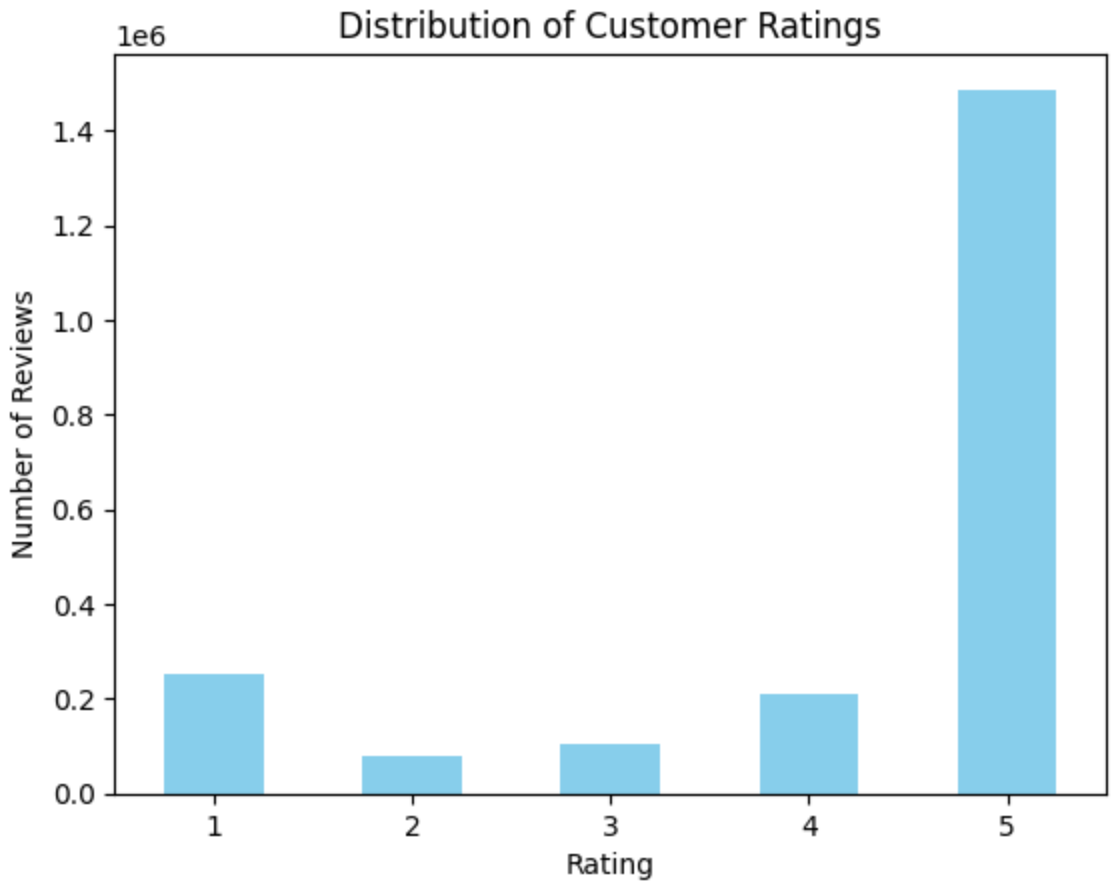
In this milestone, I began by defining the problem of classifying appliance reviews into positive or negative sentiment. I then conducted initial research to understand the problem space, reviewed relevant literature, and established my hypotheses. Here's the code I used to begin processing the data:

```
import pandas as pd
import io  # Import StringIO

chunk_size = 100000  # Process 100K rows at a time
chunks = []  # List to store processed chunks

# Read file line by line
with open(file1, "r") as f:
    batch = []  # Temporary list to store lines
    for i, line in enumerate(f):
        batch.append(line)
        if (i + 1) % chunk_size == 0:  # Every 100K rows, process the chunk
            chunk_df = pd.read_json(io.StringIO("".join(batch)), lines=True)
            chunks.append(chunk_df)
            batch = []  # Reset batch

    # Process remaining rows (if any)
    if batch:
        chunk_df = pd.read_json(io.StringIO("".join(batch)), lines=True)
        chunks.append(chunk_df)

# Combine all chunks into a full DataFrame
df_item_reviews = pd.concat(chunks, ignore_index=True)

# Read metadata file normally
df_item_metadata = pd.read_json(file2, lines=True)
```

Next, I conducted exploratory data analysis (EDA) to understand the distribution of the data and any potential patterns. For example, I visualized the distribution of ratings across all reviews:

```
# Count ratings
rating_counts = df_item_reviews['rating'].value_counts().sort_index()
```

```
# Plot bar chart
rating_counts.plot(kind='bar', color='skyblue')
plt.title('Distribution of Customer Ratings')
plt.xlabel('Rating')
plt.ylabel('Number of Reviews')
plt.xticks(rotation=0)
plt.show()
```

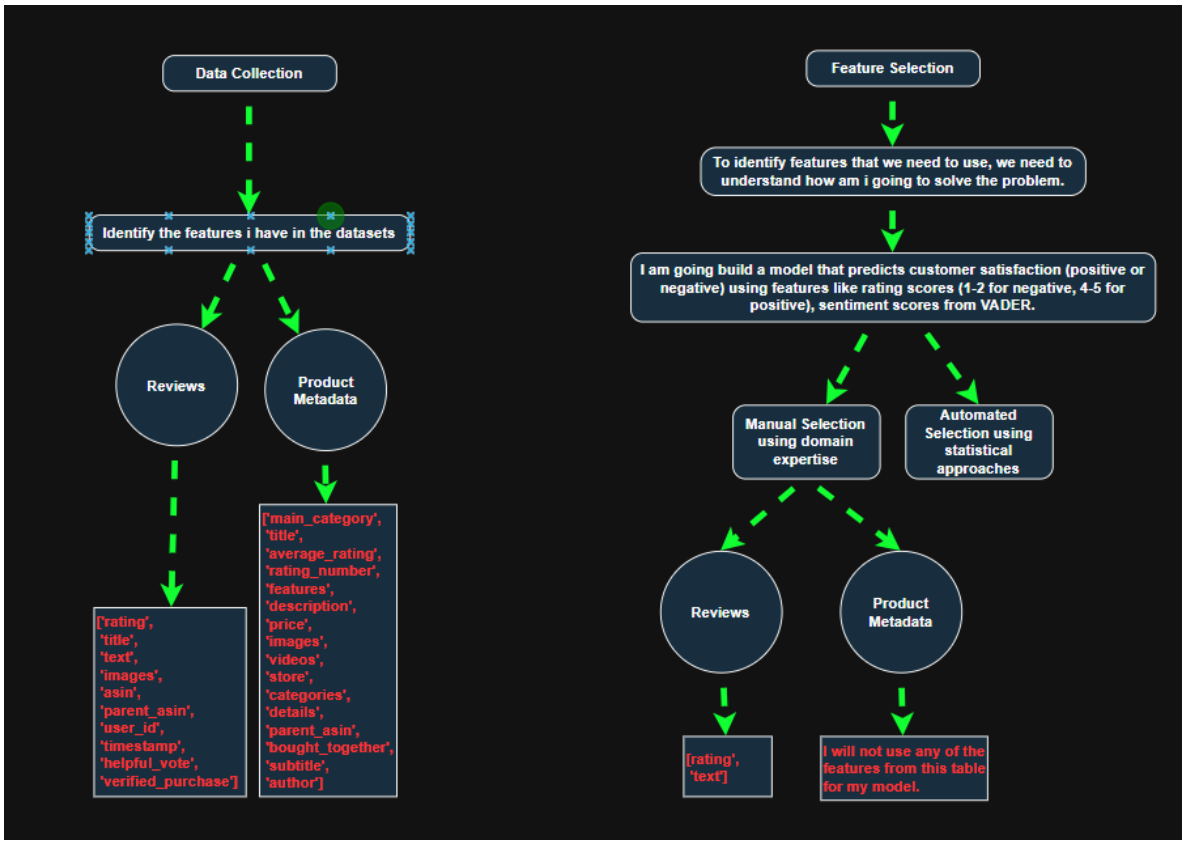


I found that most ratings were positive, while only a few products consistently received negative reviews. This observation helped me from my initial hypothesis and guide the next steps in my analysis.

The results shown in this chart are crucial because they highlight potential bias in the model selection process. In supervised machine learning, models learn from the given data, and if most ratings are high, the model may develop a bias toward predicting more positive outcomes. This imbalance could impact the model's accuracy and generalization, making it essential to account for rating distribution when training and evaluating the model.

Milestone 2: Exploratory Data Analysis (EDA)

For the second milestone, based on the stakeholder's business question, identified the essential features from each data frame.



This flowchart illustrates the features available, and based on my business question, I determined that rating and text are the only relevant features, simplifying the problem-solving process.

I conducted a more thorough data cleaning and implemented a sentiment analysis model using natural language processing (NLP). The next milestone will include these steps.

Milestone 3: Data Preparation and Model Building

In the third milestone, I moved towards building a model to predict sentiment based on the review text. This involved data preprocessing (such as tokenizing the text and converting it into numerical features) and model training. Here is the code I used for this phase:

1. **Cleaning reviews using RE module.**

```
# Function to clean the review text
def clean_review_text(text):
    text = text.lower()  # Convert to lowercase
```

```
    text = re.sub(r'[^\w\s]', '', text)  # Remove punctuation
    text = re.sub(r'\d+', '', text)  # Remove numbers
    return text

# Apply the function to clean the review text
df_logistic['cleaned_review'] =
df_logistic['review_text'].apply(clean_review_text)
```

2. **Tokenization:**

```
# using word tokenize
df_logistic['tokenized_review'] =
df_logistic['cleaned_review'].apply(word_tokenize)
```

3. **Stopwords**:

```
# Define the list of stop words from nltk
stop_words = stopwords.words('english')
df_logistic['filtered_tokenized_review'] =
df_logistic['tokenized_review'].apply(lambda tokens: [word for word in tokens if
word not in stop_words])
```

4. **Stemming/Lemmetization:**

Since my project involved analyzing reviews, I opted for lemmatization over stemming to create a more accurate bag of words.

```
# Initialize lemmatizer
lemmatizer = WordNetLemmatizer()

# Apply lemmatization
df_logistic['lemmatized_review'] =
df_logistic['filtered_tokenized_review'].apply(
    lambda tokens: [lemmatizer.lemmatize(word) for word in tokens]
)

# Join lemmatized words back into a string for text-based machine learning models
(BoW, TF-IDF)
df_logistic['final_review'] = df_logistic['lemmatized_review'].apply(lambda
tokens: ' '.join(tokens))
```

The logistic regression model was trained on the review text after converting it to a numeric representation using TF-IDF. After training, I evaluated the model using precision, recall, and F1-score.

```
X = df_logistic['final_review']
y = df_logistic['sentiment']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
```

I then fit tf-idf to the training set

```
# Create TF-IDF Matrix
tfidf_vectorizer = TfidfVectorizer()

# Generate TF-IDF matrix
tfidf_matrix_train = tfidf_vectorizer.fit_transform(X_train)

# Display TF-IDF Matrix Dimensions
print("TF-IDF Matrix Dimensions:", tfidf_matrix_train.shape)
```

Applied the tf-idf to the test set

```
tfidf_matrix_test = tfidf_vectorizer.transform(X_test)
print("TF-IDF Matrix Dimensions:", tfidf_matrix_test.shape)
```

trained logistic regression and fit on the training set

```
logit = LogisticRegression()
logit.fit(tfidf_matrix_train, y_train)
logit.score(tfidf_matrix_test, y_test)
```

Conclusion

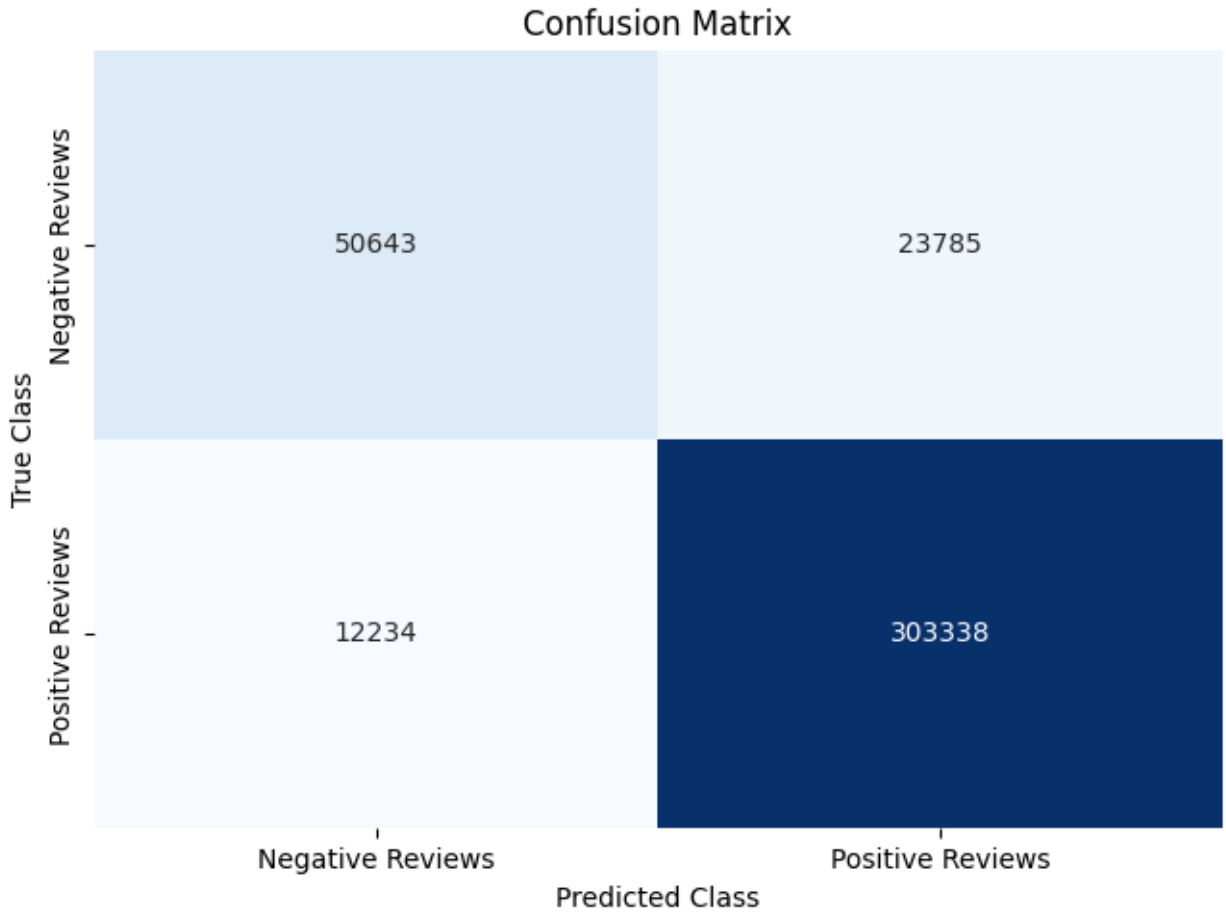
Model Evaluation and Insights

The model achieved an accuracy of 0.91, with a detailed report below highlighting the F1 scores for both negative and positive reviews.

```
Logistic Regression Report:

                   precision    recall  f1-score   support

Negative Reviews       0.81      0.68      0.74     74428
Positive Reviews       0.93      0.96      0.94    315572

        accuracy                           0.91    390000
       macro avg       0.87      0.82      0.84    390000
    weighted avg       0.90      0.91      0.90    390000
```

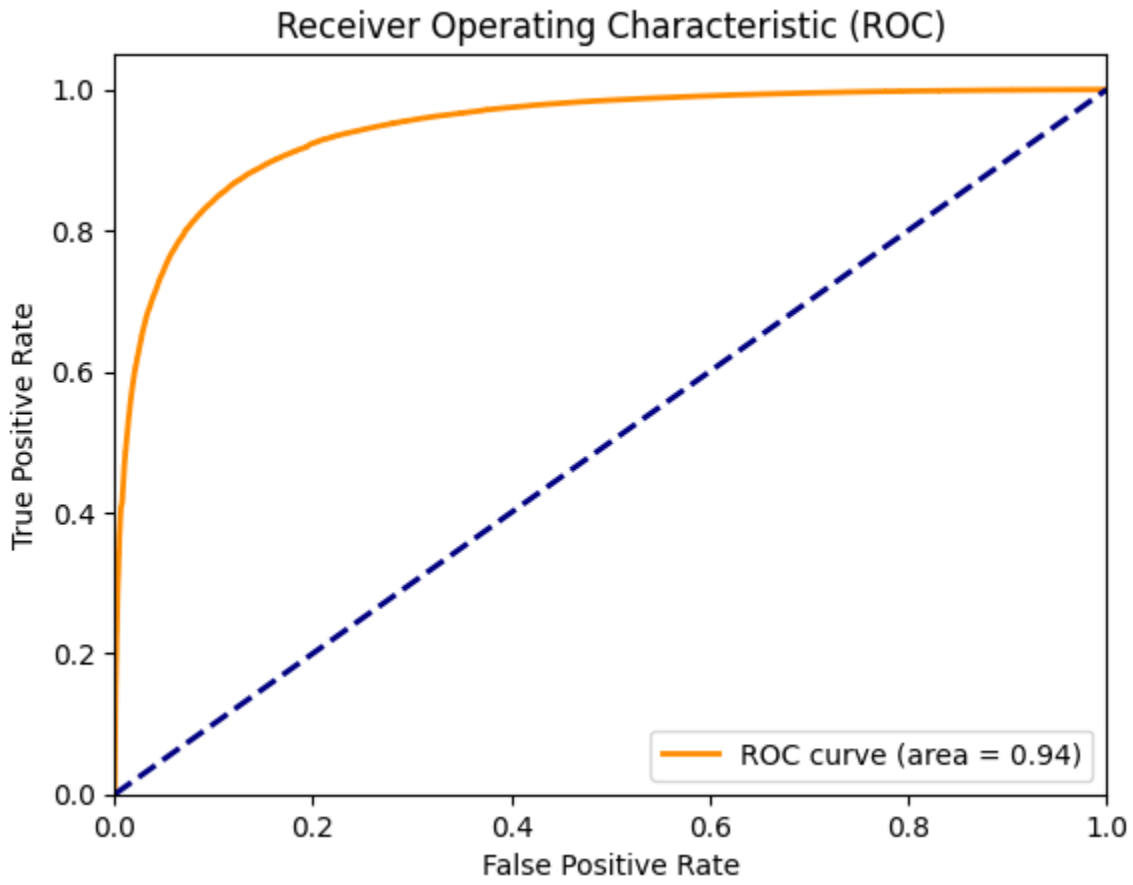
An F1 score of 94% for positive reviews and 74% for negative reviews indicates a performance imbalance. As previously mentioned, the model is biased due to most reviews being positive.

A confusion matrix representation to showcase where my model did well and bad.



- **True Negative (Top-Left):** 50,643 reviews were correctly classified as negative.
- **False Positive (Top-Right):** 23,785 negative reviews were misclassified as positive.
- **False Negative (Bottom-Left):** 12,234 positive reviews were misclassified as negative.
- **True Positive (Bottom-Right):** 303,338 reviews were correctly classified as positive.

The model performs well, particularly in identifying positive reviews, but has some misclassification of negative reviews.



The logistic regression model has 91% accuracy, performing well overall. It excels at identifying positive reviews but shows moderate performance with negative reviews (80% precision, 67% recall). The weighted averages for precision, recall, and F1-score are all 0.90, reflecting strong overall performance. The Area Under the Curve (AUC) is 0.94, indicating a strong classifier.

After completing the project, the Logistic Regression model performed well on classifying appliance review sentiments. The evaluation metrics showed that the model was able to distinguish between positive and negative sentiments with reasonable accuracy (91%).

Deployment Readiness

The model isn't fully ready for deployment yet. To improve these topics can be addressed:

- **Addressing Bias**: Adjust class weights or resample the data to handle class imbalance, especially for negative reviews.
- **Training on More Data**: Add more labeled negative reviews to improve recall.

- **Threshold Adjustment**: Adjust classification thresholds to improve recall for negative reviews.

Recommendations for Future Work

In the future, I recommend exploring more advanced NLP techniques such as BERT, which can capture contextual meaning in the reviews better than simpler models. I also suggest applying the model to different product categories and continuously updating it to handle new data and evolving language patterns.

Challenges and Opportunities

A significant challenge I faced was handling the imbalance between positive and negative sentiment in the dataset. However, this project provides an opportunity to explore deeper into NLP and build models that can address more nuanced sentiment categories or multi-class classifications.